Visual attention strategies for target object detection

by

Ibrahim M. H. Rahman

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
2018

Abstract

The *human visual attention system* (HVA) encompasses a set of interconnected neurological modules that are responsible for analyzing visual stimuli by attending to those regions that are salient. Two contrasting biological mechanisms exist in the HVA systems; bottom-up, data-driven attention and top-down, task-driven attention. The former is mostly responsible for low-level instinctive behaviors, while the latter is responsible for performing complex visual tasks such as target object detection.

Very few computational models have been proposed to model top-down attention, mainly due to three reasons. The first is that the functionality of top-down process involves many influential factors. The second reason is that there is a diversity in top-down responses from task to task. Finally, many biological aspects of the top-down process are not well understood yet.

For the above reasons, it is difficult to come up with a generalized top-down model that could be applied to all high level visual tasks. Instead, this thesis addresses some outstanding issues in modelling top-down attention for one particular task, *target object detection*. Target object detection is an essential step for analyzing images to further perform complex visual tasks. Target object detection has not been investigated thoroughly when modelling top-down saliency and hence, constitutes the may domain application for this thesis.

The thesis will investigate methods to model top-down attention through various high-level data acquired from images. Furthermore, the thesis will investigate different strategies to dynamically combine bottom-up and top-down processes to improve the detection accuracy, as well as

the computational efficiency of the existing and new visual attention models. The following techniques and approaches are proposed to address the outstanding issues in modelling top-down saliency:

1. *A top-down saliency model that weights low-level attentional features through contextual knowledge of a scene*. The proposed model assigns weights to features of a novel image by extracting a contextual descriptor of the image. The contextual descriptor plays the role of tuning the weighting of low-level features to maximize detection accuracy. By incorporating context into the feature weighting mechanism we improve the quality of the assigned weights to these features.

2. *Two modules of target features combined with contextual weighting to improve detection accuracy of the target object*. In this proposed model, two sets of attentional feature weights are learned, one through context and the other through target features. When both sources of knowledge are used to model top-down attention, a drastic increase in detection accuracy is achieved in images with complex backgrounds and a variety of target objects.

3. *A top-down and bottom-up attention combination model based on feature interaction*. This model provides a dynamic way for combining both processes by formulating the problem as feature selection. The feature selection exploits the interaction between these features, yielding a robust set of features that would maximize both the detection accuracy and the overall efficiency of the system.

4. *A feature map quality score estimation model that is able to accurately predict the detection accuracy score of any previously novel feature map without the need of groundtruth data*. The model extracts various local, global, geometrical and statistical characteristic features from a feature map. These characteristics guide a regression model to estimate the quality of a novel map.

5. *A dynamic feature integration framework for combining bottom-up and top-down saliencies at runtime*. If the estimation model is able to predict the quality score of any novel feature map accurately, then it is possible to perform dynamic feature map integration based on the estimated value. We propose two frameworks for feature map integration using the estimation model. The proposed integration framework achieves higher human fixation prediction accuracy with minimum number of feature maps than that achieved by combining all feature maps.

The proposed works in this thesis provide new directions in modelling top-down saliency for target object detection. In addition, dynamic approaches for top-down and bottom-up combination show considerable improvements over existing approaches in both efficiency and accuracy.

List of Publications

Parts of the research work of this thesis have been published/submitted in the following conferences and journals:

1. Ibrahim Rahman, Christopher Hollitt, and Mengjie Zhang, "Feature map quality score estimation through regression", *IEEE Transactions on image processing*, DOI: 10.1109/TIP.2017.2785623.

2. Ibrahim Rahman, Christopher Hollitt, and Mengjie Zhang, "Task Driven Feature Selection for Top-down Visual Attention", *Information Sciences*, 2017, *Conditionally accepted*.

3. Ibrahim Rahman, Christopher Hollitt, and Mengjie Zhang, "A dynamic feature map integration approach for predicting human fixation", *in proc. of the International Conference on Image and Vision Computing New Zealand (IVCNZ)*, Palmerston North, New Zealand, November 2016.

4. Ibrahim Rahman, Christopher Hollitt, and Mengjie Zhang, "Contextual-based top-down saliency feature weighting for target detection", *Machine Vision and Applications*, vol. 27, no. 6, pp. 893 – 914, August 2016.

5. Ibrahim Rahman, Christopher Hollitt, and Mengjie Zhang, "Information Divergence Based Saliency Detection with a Global Center-Surround Mechanism", *in proc. of the International Conference on Pat-*

tern Recognition (ICPR), Stockholm, Sweden, August 2014, pp. 3428 – 3433.

Acknowledgments

All praise unto ALLAH (The GOD), Who made it possible to complete my thesis work. I would also deeply like to thank my dear parents for their continuous prayers for me, which they have always done since I was born. Particularly, I would like to mention my mother, who passed away during my PhD. She always encouraged me for higher studies and was her dream for me to acquire a PhD degree. I hope that I would fulfill her wish. I would also like to thank all my other family members (brothers and sisters) for their prayer and continuous support.

I acknowledge the support of my beloved wife both mentally and financially at each stage of my PhD. I also thank her for taking care of my two lovely daughters, and providing me recently with another gift in the form of a lovely baby boy. Thank you *Aisha* (my wife).

I am very thankful to my supervisors, Dr. Christopher Hollitt (Chris) and Prof. Mengjie Zhang (Meng) for their continuous support. Because of them, I have improved my research thinking, knowledge of the research area and particularly academic writing. I had plenty of academic discussions with them that benefited me in developing my research skills. I would also like to thank them for spending and dedicating their precious time in reading and suggesting valuable changes in my research papers and thesis, which I believe is a very hard work.

I am also very grateful to Dr. Christopher Hollitt for arranging a research grant for me to work as a research assistant after submitting my thesis. In addition, I also appreciate his approach is discussing each aspect of my research in detail so that I should be aware of my own work in depth and with full understanding. I have a good understanding of the visual attention system with his expert feedbacks and discussions on this topic. Thank you Chris for your time and effort.

I would also like to thank Prof. Mengjie Zhang for his suggestions in critical moments of my PhD journey. I would like to acknowledge his expertise in the field of machine learning and evolutionary computation in particular. Although,I have not fully explored the evolutionary computation field, but with the techniques I used in my thesis work, I had excellent feedback from him. Thank you Meng for giving me time, despite being very busy.

I am also thankful to Victoria University of Wellington for awarding me the Victoria Doctoral Scholarship and helping me through hardship funds. I also appreciate the university for providing me with all necessary resources and research materials that enabled a smooth PhD journey.

Finally, I would like to dedicate this thesis to both my *late* beloved mother (*Rafiul-nisa Hussain*) and my father (*Mohammad Hussain*).

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Chapter introduction

Humans have a remarkable ability to handle various complex vision problems effectively and efficiently through a process called *visual attention*. Visual attention is a cognitive process that attempts to sample and analyze certain regions of a visual field while ignoring irrelevant areas [4]. Initial studies in human vision suggest the existence of a fast, data-driven, bottom-up (BU) influence and a slower, task-specific, top-down (TD) influence that contribute collectively to guiding visual attention [5]. The active involvement of both influences results in a more effective attention strategy and efficient saccadic movement. Saccade is defined as the process of transferring the fixation from one point to another in a certain pattern [5].

More recently, researchers have devoted their efforts in modelling visual attention to proposing various bottom-up and top-down computational models that mimic some of the functionalities of a human visual attention system. These models have been widely used in many real-time applications such as but not limited to object recognition [6–8], medical images analysis [9], image compression [10], object tracking [11] and target object detection [12].

Most of the work is dedicated towards modeling BU influence, while fewer contributions have been made to model the TD influence and how to combine it with BU influence to maximize both effectiveness and efficiency of a specified visual task.

As a result of neurobiological studies, it has been hypothesized that most of the complex visual tasks performed by humans are driven by high-level top-down signals generated from certain areas of the visual cortex that are dependent on the nature of the task itself [5]. Furthermore, the nature of such signals and the factors controlling the initiation of those signals are not completely known. Hence, it is very challenging to devise a universal computational model for explaining the nature of such top-down influence for each visual task.

An even more challenging research question is how both TD and BU influences are combined during a complex visual task. What role does the BU play in such a high-level task? Since the BU process is responsible for detecting regions of the images that are deemed interesting (in visual attention literature these are also known as salient regions), its contribution in more demanding scenarios such as target object recognition is not elusive.

Hence, the main goals of this thesis are twofold. The first is to propose a TD saliency model effective for one specific visual task: target object detection. We define a *target object* as an object of interest to be searched for in an image where the objective is simply to locate that object. Hence, the proposed TD saliency model would effectively detect the target object with high accuracy and a minimum number of saccadic movements. Secondly, we propose a computational model that will combine the BU process with the TD process to maximize the detection of the target object.

1.2 Scope

An active research area in visual attention is to describe sampling strategies of a visual field under the influence of a task [13–15]. Because of the diversity in visual tasks, research in this area has not yielded a generalized model. One particular task that has been focused upon more often is a guided search for target object detection [16, 17], which is also the focus of this research work. It is well known that while in a guided search mode, a variety of processing takes place before detecting the target object [18]. Only those candidate objects or regions are sampled that exhibit pertinent visual characteristic with respect to target object.

Previously, modelling of task-driven visual attention has been studied from three perspectives: the features required to encode a task, the methodology and formulation of the task-driven attention and finally the interaction between data-driven and task-driven attention factors.

Current implementations of task-driven visual attention rely on very high level computationally expensive features [19]. In addition, the performance of these methods degrades when the target object is present in complex scenes with cluttered background and distracting objects. Furthermore, among these implementations, very few models combine both task-driven and data-driven process to boost the efficiency and efficacy of the attention system.

To tackle the issues presented above, "intelligent" approaches are needed to build a generalized visual attention system that aims at detecting a generic target object. This needs to maintain a balance between detection accuracy and computational efficiency to be feasible for practical purposes.

1.3 Motivations

There exist several biological plausible data-driven models and techniques (also known as saliency or bottom-up (BU) based approaches) that are directly or indirectly inspired by cognitive concepts. One of the leading works in biologically inspired bottom-up saliency is by Itti et al. [1]. This model has been the basis of later models and the standard benchmark for comparison. The model begins by extracting low level features followed by a centre-surround normalization mechanism. This process results in generating feature maps. These maps are then integrated and normalized at different scales to yield the conspicuity. Finally, the conspicuity maps are combined linearly to build the final saliency map.

1.3.1 Top-down saliency modelling through feature weighting

One category of task-driven (also known as top-down (TD)) modelling is performed by appropriately weighting these bottom-up *Itti* features [20–26]. This type of modeling is inspired by the cognitive fact that low-level signals generated by a fast bottom-up process are amplified by stimulating signals that are triggered when performing a task [27]. In previous approaches, these weights were computed or learned without considering high-level knowledge of the target, background, distractors and other prior knowledge of the scene. One such critical high-level information contained in images is the context. Contextual information provides a holistic description of the contents on an image.

For instance, when searching for a red ball within an image containing circular patterns, the colour should have a higher weight than orientation in order to detect it. However, the same is not true when searching for the same object in an image having rectangular red background patterns. As a result, when learning weights without considering high-level visual structure of a scene, the approach fails to adapt to scenes with contextual variation. As a result, a novel dynamic contextual based bottom-up feature weighting mechanism is needed to model TD attention and improve the accuracy of detecting the target object.

Another aspect of modeling TD influence through weighting is the set of features used for generating the final saliency map. The basic model of *Itti* which is used as a framework for feature weighing utilize three primitive features of colour, intensity, and orientation. For instance, a model proposed by Frintrop [20] called *VOCUS* learns the weights of these primitive features by taking the ratio of the average saliency of the most salient region and the saliency of the rest of the regions. Since the model uses only primitive features, it only predicts regions that are likely to contain the target. As an extension to their work, an additional step of classification was introduced where high-level descriptors were used to learn classifiers that are tuned to detect the target object under consideration [21].

1.3.2 High-level features in top-down saliency

In many extensions to the basic model, other high-level features such as text [28], face [28] and motion [29] are used to complement the basic features, mostly to predict human fixation. In majority of cases, these features are computationally expensive and require longer processing time than the low-level features, thus affecting the efficiency of the model.

This indicates that modeling TD saliency by weighting the currently used primitive features has limitations in detecting the target object. Hence, a richer set of various low to mid-level features that are both computationally efficient and useful for generic target object detection is required to model TD influence through weighting to avoid the use of high level computationally demanding features.

In some cases, it becomes inevitable to use target specific features complemented by various low-level bottom-up features (e.g., face detector along with basic *Itti* features [28]) when searching for faces. Target specific features are a priori more likely to be effective in searching for the target than low-level features. Bottom-up features collectively produce a map that highlights interesting regions of an image while target specific TD features yield a map that identifies regions of interest with respect to the target object. When both are combined, a final saliency map is generated that combines both detections.

1.3.3 Top-down and bottom-up saliency integration

Another research direction in modelling TD saliency is therefore how and when to combine it with BU saliency to improve the detection. In high-level visual tasks, it could be argued that TD features have the major role in detecting the target object [30]. However, this is not always true, as low-level BU features can sometimes identify regions of the image that may contain the target object more effectively than those extracted by target specific TD features [31]. Similarly, low level BU features might effectively eliminate potential targets identified by a more complex TD feature.

Current techniques that attempt to combine both saliency processes lack a dynamic combination strategy. The two saliency processes are either combined statically [20, 22, 24] or through a pre-defined spatio-temporal function that weighs TD process more that BU [23]. A major disadvantage of such approaches is that no information regarding the interrelation between both the two processes is available.

It is possible that certain BU features do not contribute directly to target detection, but when combining them with certain TD features could lead to an improvement in the detection accuracy. In addition, depending on the contextual information of individual images, the level of interaction between the features of both types vary.

A dynamic framework for combining both influences as well as avoiding computation of irrelevant features belonging to either of the two influences is needed to understand such interactions. The dynamic strategy allows irrelevant features to be removed on an image by image basis. The inclusion of unimportant or redundant features has two main disadvantages. First, the efficiency degrades because of the extra computation of such features, and secondly, the detection performance could degrade due to too large search space. A dynamic solution to this problem could be addressed by formulating the problem as a feature selection through optimization in such a manner that all the features belonging to either TD or BU are pooled together for the selection process.

When TD and BU features are combined through the feature selection process by either certain optimization criteria or just through a static combination process, the result is assumed to work on all images within a dataset. However, the performance could vary from image to another. For instance, Ehinger et al. [3] suggested that to maximize detection of pedestrians in a dataset of outdoor images, three types of features need to be combined through a visual attention setup for all the images. Although the achieved results supported their hypothesis, the optimum set of features varied from image to image and was not exploited in their work.

A dynamic mechanism is needed to combine various features on an image basis. Several techniques have been proposed in the past to handle this problem [32]. These techniques select or ignore a feature from the combination process on-line. Whenever a feature is computed (either belonging to TD or BU), and before the final integration, an intermediate map is generated which is called the *feature map*. The feature map highlights activation points deeming salient in an image according to that feature. By inspecting the visual characteristic of the feature map, a decision is made to whether to keep or drop that feature from the combination process. Depending on the properties of such feature maps, it will be either selected or ignored for the final combination process.

Previous methods of feature map selection are highly dependent on the particular measure of compactness and its variant [32, 33]. Feature maps that exhibit high compactness score are favored to go into the combination process. For salient object detection, compactness is typically found to be very effective in measuring a quality score of a map [34]. However, this measure might not work when considering feature maps for fixation as they exhibit visual characteristic that are considerably different from salient object detection feature maps. In addition, compactness scores are computed directly from the map itself and no intelligent learning is involved, which makes such approaches rather static.

A new approach is needed that can precisely estimate a quality measure of fixation feature maps. This is possible by extracting useful information from the feature maps. The information should describe the visual attributes of such maps. Furthermore, for any novel feature map, in order to dynamically estimate how good it is for target object detection (measured through some quality score), a model is needed that would associate the visual attributes to a quality score value. If this is achieved accurately, it would be possible to combine only good quality feature maps dynamically on individual image basis.

1.4 Thesis Statement

The main theme of this thesis is to model the task-driven cognitive influence and propose machine learning approaches by which it is integrated with data-driven influence to maximize both accuracy and efficiency of the system in detecting target objects.

1.5 Goals

The goal of this thesis is to primarily develop a generalized visual attention system that incorporates both task-driven and data-driven saliency factors for detecting a generic target object. The thesis does not aim at mimicking human visual attention model, but rather to come up with computational attention models that achieve the desired goal of target object detection effectively and efficiently. There are a number of issues that will be addressed in this thesis to achieve the overall goal. The main goal is divided into the following objectives:

1. To develop a generalized framework of task-driven influence that incorporates high-level knowledge of the scene into the system for better generalization over images with diverse visual content. In order to achieve this objective, the following tasks are performed:

  - Model task-driven influence through weighting of low-level features that are tuned for a particular target object. For the model to generalize over a variety of images containing the target object, a scene context element needs to be introduced that is used to dynamically assign appropriate weights to the features.
  - Compare the proposed contextual based feature weighting visual attention model with traditional attention models that perform plain weighting of features without incorporating high-level knowledge into the system.

2. To explore various low to mid-level features to be added to the existing feature weighting based TD models that previously utilized only primitive set of features. Expanding the set of low-level features will increase the effectiveness of the model in describing the target object without the need for high-level target specific detectors.

3. To incorporate knowledge of the target into the contextual based TD saliency weighting. This is to be done by only considering low-level features without opting for high-level target specific feature for an efficient attention system.

4. To design an approach that combines good features from both TD and BU influences for maximizing the detection accuracy over a dataset of images. This requires formulating the problem as feature selection problem through optimizing an objective function to select important and relevant features from both categories. Such a design will enable features from both categories to interact with in order to maximize target detection accuracy.

5. To develop a mechanism through which a feature map is selected or removed from the feature map integration process to perform a dynamic feature map integration on individual image basis. This will improve the efficiency and boost the detection accuracy of the

model. The implementation of this approach will consist of the following tasks:

- Extract efficiently computed characteristic features from the feature map that describes its visual attributes. The features gather geometrical, local, global, statistical and spatial information of the feature map.
- Learn a regression model that estimates a utility score of any feature map based on the characteristic features extracted from these maps. A decision on the selection or exclusion of the feature map from the integration process is made based on the estimated quality score.

1.6 Major contributions

By achieving the above objectives, this thesis will provide the following major contributions in modeling task-driven saliency for target object detection in the field of visual attention:

1. A new TD feature weighting saliency model is proposed that learns feature weights through the contextual information within an individual image. This model provides a dynamic mechanism to tune the feature weights based on the context of a novel image. This modelling will show the superiority of incorporating contextual information in tuning the feature weights over those models that do not use contextual information while learning such weights.

   The model includes additional low-level features apart from the *Itti* features. These features demonstrate that a better target detection accuracy is achieved than those primitive low-level features commonly used in previous TD attention models. The features are computationally efficient and a marginal increase in processing time is observed.

Part of the research work in this chapter is published in the following conference and journal:

- Ibrahim Rahman, Christopher Hollitt, and Mengjie Zhang, "Information Divergence Based Saliency Detection with a Global Center-Surround Mechanism", *in proc. of the International Conference on Pattern Recognition (ICPR)*, Stockholm, Sweden, August 2014, pp. 3428 – 3433.
- Ibrahim Rahman, Christopher Hollitt, and Mengjie Zhang, "Contextual-based top-down saliency feature weighting for target detection", *Machine Vision and Applications*, vol. 27, no. 6, pp. 893 – 914, August 2016.

2. A new approach is proposed that combines target information through low-level features with the contextual information. In our knowledge, this is the first model that combines the contextual information, target feature and a recognizer that is tuned for a particular target object. The model is based on attentional modules that learn two separate weight vectors which are tuned by the contextual information and the target object information. We show that the detection accuracy in the form of F-measure score is always boosted when incorporating target information (in the form of target features or a target tuned recognizer). The model is tested and analyzed on seven challenging datasets with 12 different objects contained in complex and cluttered background scenes to establish the merits of the proposed approach. In addition, the model is very efficient as it only utilizes low-level features in performing target object detection.

3. A new approach of combining TD and BU influences by exploiting the interaction between various features from both categories is proposed. A novel formulation of the problem is performed as feature selection using particle swarm optimization (PSO). The features from

both categories are pooled into the same set from which the selection takes place, rather than by considering them as separate channels. The obtained results show that the proposed model outperform state-of-the-art saliency techniques in combining both processes for detecting the target object. Furthermore, the model provides feature importance profiles that are interpretable as they describe the features that are useful from both categories and those which are irrelevant to the task.

Part of the research work in this chapter is under revision after passing the revision round in the following journal:

- Ibrahim Rahman, Christopher Hollitt, and Mengjie Zhang, ”Task Driven Feature Selection for Top-down Visual Attention”, *Information Sciences*, 2017, *Conditionally accepted*.

4. A new dynamic feature map integration strategy is proposed. The proposed approach is based on estimating a quality score of a fixation feature map through regression from a set of characteristic features that are used to visually describe a fixation feature map. The following contributions are made:

   - A new set of 29 features are proposed that extract geometrical, statistical, local and global information from a feature map and would describe the visual appearance of the map. The features are very computationally efficient and can be used in runtime applications.

   - A new approach for estimating the quality score of a feature map is proposed based on a regression problem. The proposed random forest regressor learns a model from the features extracted from a feature map (called characteristic features) to estimate a quality score of a novel feature map accurately without the need of the groundtruth data.

- Two new feature map integration frameworks are proposed that utilizes the estimation model for a dynamic feature map integration. By integrating our proposed approach with a two simple proposed feature map integration frameworks, we demonstrate that our model achieves a higher human fixation prediction accuracy and efficiency than that achieved by combining all feature maps.

Part of the research work in this chapter is published in a conference proceeding and another under revision after passing the first revision round in a journal:

- Ibrahim Rahman, Christopher Hollitt, and Mengjie Zhang, "A dynamic feature map integration approach for predicting human fixation", *in proc. of the International Conference on Image and Vision Computing New Zealand (IVCNZ)*, Palmerston North, New Zealand, November 2016.
- Ibrahim Rahman, Christopher Hollitt, and Mengjie Zhang, "A feature map quality score estimation through regression", *IEEE Transactions on Image Processing, Submitted*.

1.7 Thesis Organization

The remainder of the thesis is organized as follows. Chapter 2 presents a background discussion on visual attention systems with emphasis on task-driven attention modeling. Chapters 3 to 6 establish the main contributions of the thesis. Finally, the thesis conclusion and potential future extensions are presented in chapter 7.

In chapter 2, various attention systems are discussed with emphasis on TD saliency. Previous works related to attention system combination, feature selection and proto-object models for target object detections and their shortcomings are highlighted in detail.

To perform a more controlled weighting of low-level features to model TD saliency for target detection, chapter 3 introduces a novel approach that incorporates contextual information into the weighting mechanism to model TD saliency for target object detection. In addition, various efficiently computed features are discussed in the chapter to address object detection generalization through low-level features without using target specific features. A thorough analysis performed on different challenging datasets and comparison with various state-of-the-art techniques is presented in the chapter.

To see how target information can be incorporated into a contextual based TD saliency model, chapter 4 provides a mechanism to combine both sources of information to maximize the detection accuracy of the target object. The chapter also discusses how to utilize a target tuned recognizer to boost the detection accuracy of the context/target attention model.

Chapter 5 introduces a generalized framework that combines important and relevant features from both TD and BU saliencies for target object detection. The framework utilizes PSO capability to effectively perform feature selection from a pool of features consisting of both TD and BU features. The objective function represents the agreement between the final produced saliency map through some feature combination and the ground-truth maps over the entire training dataset. A comparison with state-of-the-art TD attention models is performed both in terms of detection accuracy and speed of detection that is established by the number of fixations required to reach to the target object.

A feature map quality score estimation approach is proposed in chapter 6. The model consists of extracting visual characteristic features from a feature map to describe its visual attributes. Furthermore, a regression model that estimates a quality score of any novel feature map through the characteristic features is explained in this chapter. Finally, two dynamic approaches to incorporate the quality estimation model and a feature map integration framework are presented in this chapter to predict where human fixate on when searching for pedestrians. The results are compared with a popular fixation feature map combination strategy and a comprehensive analysis and observations are provided in the chapter.

Finally, chapter 7 summarizes the thesis objectives and how they have been achieved through the proposed models. Furthermore, thesis contributions are highlighted along with possible future works in this domain.

Chapter 2

Literature Survey

2.1 Chapter introduction

This chapter provides an overview of the biological working of human visual attention system and the existing computational saliency models. The chapter is divided into two parts. The first part provides a broad spectrum of topics related to active vision, particularly visual attention. For a better understanding of the concept of saliency associated with visual attention, a description of bottom-up saliency along with a brief discussion on various computational models of bottom-up saliency is presented. The chapter also concentrates on one particular application of active vision systems which is object detection (or target object detection). Because the thesis uses machine learning techniques to achieve the overall goal, an overview of machine learning concepts, parameters, tasks being used for and paradigms are discussed briefly. As part of machine learning, the feature selection process is also discussed here, as some objectives of the thesis coincide with it.

The second part of the chapter highlights those topics required to understand the direction of the research work of this thesis. A thorough discussion on top-down saliency and its relevance to high-level tasks is given in this chapter. This discussion also highlights the limitations of the existing saliency methods in successfully performing high level visual tasks such as target object detection. Finally, the chapter also discusses various approaches and their limitations to combine both types of saliency to maximize the detection accuracy of target objects.

2.2 Background

The goal of this chapter is to provide a generic information about some areas of active vision and machine learning relevant to the thesis topic.

2.2.1 Active vision

In the machine vision domain, active vision describes systems that are capable of dynamically manipulating the viewpoints of scene acquisition devices [34, 35]. Unlike passive computer vision techniques, where there is no control over the data acquisition process, the active vision systems exhibit a non-trivial degree of decision making during the image acquisition process. Passive vision techniques do not account for dynamic nature of the real world scene and the problems associated with it, whereas active vision techniques have a great ability to cope with such challenges [35].

An astonishing active vision system in existence that can perform complex vision tasks simultaneously is the visual attention system in primates, particularly human visual attention (HVA). Human vision has a remarkable ability to handle various complex vision problems effectively and efficiently through a process called visual attention [13]. Visual attention is a cognitive process that attempts to sample and analyze a visual field by attending to certain parts of the visual area through a selective and guided search mechanism [4]. It seeks to optimally schedule sensory resources of the neurological system to areas of an input that are deemed most important. Subsequent attention is then preferentially devoted to those areas, whether it is in the form of further observations, or ongoing analysis to better determine the nature of the image [36].

Nearly all active vision techniques try to mimic the functionality of human visual attention system. These techniques have been applied to a broad range of real-world problems in robotics, military, education and security. Hence, in order to better understand the human visual attention system, the next section provides a discussion on biological components of HVA and their functionality.

2.2.2 The human visual attention (HVA) system

Neurobiologists, psychologists, and engineers have investigated the working of visual attention system in primates and specifically humans from different perspectives. These studies have resulted in a better fundamental understanding of the system, along with considerable applications in various fields. However, many secrets of this system can not yet be explained or remain to be explored.

There are two directions in the literature concerning HVA, one followed by the neurobiologists and psychologists that constitutes the field of Computational Neuroscience which deals with the theoretical study of the brain and its functionality and association with vision. The other direction is the development of computational models of the HVA. The term biological HVA and computational HVA will be used to describe these directions respectively. There are many good literature surveys that cover detailed aspects of both directions such as [4,37] for biological HVA and [18,19,38] for the computational modelling of HVA.

2.2.2.1 Biological attention

This section will give a brief overview of how the visual information processing takes place inside the HVA system [18].

Retina:

The retina reduces the huge amount of information coming from the external sources through a linear operation called the centre-surround mechanism. The retinal ganglion cells are responsible for performing this operation through specialized receptive fields called ON cells and OFF cells. The ON cells increment the light intensity in the centre of the receptive field and OFF cells decreases it. Most of the computational models replicate this mechanism through a linear The difference of Gaussian (DoG) operation.

Lateral Geniculate Nucleus (LGN):
The LGN receives signals directly from retinal ganglion cells via the optic tract and from the Reticular Activating System (RAS). The RAS is responsible for arousal control and sleep-wake transitions. The LGN directs the incoming signals to the V1 and V4 visual cortex (explained next). However, the precise function of the LGN is relatively poorly understood.

Visual Cortex:
The visual cortex is the part of the cortex responsible for processing visual information. It is located in the occipital lobe and accounts for a large proportion of the human brain. It contains many sub-areas and can be grouped into two main parts, the primary visual cortex or V1 area, and the Extrastriate visual cortex such as V2, V3, V4 and V5. The functionality of each sub-area is given below:

Primary visual cortex (V1): V1 is the best-studied area of the visual system. It has the ability to process information about static objects and perform pattern recognition. The V1 neurons are tuned to extract specific features from the input information such as visual orientations, spatial frequencies, and colours. However, the initial responses of V1 consist of sets of selective spatio-temporal cells equivalent to Gabor transform and is responsible for extracting many spatio-temporal features.

These cells (sometimes called edge detectors) are sensitive to various orientations. When a light of certain orientation is projected onto the retina cells, the signals from these cells are forwarded to the V1 cells through the *thalamus* layer. Only those V1 cells that exhibit a high response to that specific orientations are activated. More interestingly, researchers found that the number of such cells and their wiring are very sensitive to past experience [18]. For instance, a subject exposed to vertical patterns more often than horizontal patterns is more likely to have more vertical cells in the V1 area than horizontal cells. Hence, these cells tune themselves with experience.

The V2 area: This area exhibits feedforward connections that deliver the signals from V1 area to V3, V4 and V5 areas. The cells contained in V2 area act as storage points and have the ability to convert short-term memory into long-term [39].

The V3 area: V3 is located in front of the V2 area. It receives the inputs from V1 and V2 and projecting them to the Posterior Parietal (PP) cortex. Some studies indicate that the area is responsible for the handling of global motion covering large portions of the visual field.

The V4 area: The exact functionality of the V4 remains a question, however, some studies suggest that it has similar functionality as V1 but can also handle complex features such as geometric shapes.

The V5 area: Also known as MT region of the Extrastriate visual cortex in humans, plays an important role in the perception of motion and orientations of some eye movements. It is also responsible for the integration of local motion signals into global perceptions. V1 provides the most important input to the MT. However, recent studies show that MT is capable of responding to various visual information even after V1 is disabled, emphasizing that there are other specialized signals entering this region before even entering the V1 region.

2.2.2.2 Functionality of the visual attention system

Out of the huge visual information bombarding the human retina, only a small region of the scene being focused on is processed. We say that the focused region has been a candidate for attention. Anzai defined attention as ”the mental ability to select stimuli, responses, memories, or thoughts that are behaviorally relevant among many others that are behaviorally irrelevant” [39]. This attention is derived by two factors, namely, the *bottom-up* process and the *top-down* process. The bottom-up process, also known as a data-driven process, is derived solely from the input stimuli. This process leads to a focus of attention on regions having features different from other parts of a scene. These regions are known as *salient* regions. Many researchers have concluded that bottom-up influence is not voluntarily suppressible and highly salient regions, therefore, capture the focus of attention (FOA) regardless of any other factors involved [40].

The top-down or task-driven process is a slow goal and task oriented mechanism performed by parts of the intraparietal cortex and superior frontal cortex. A more detailed discussion on this influence is covered in section 2.3.2.

It is important to note that while bottom-up saliency and attention have been thoroughly investigated at biological and computational levels, this is not the case for top-down saliency. One reason might be that the data-driven factors are easier to control and understand than cognitive top-down factors. In addition, bottom-up factors are used in either studying the ability of humans to detect salient regions/objects or to study where humans fixate their eye during a random search within a scene. A more challenging study which has not been explored as broadly is how humans perform high-level recognition of scenes and objects through knowledge and goals which are a topic of top-down influence on the attention for detection and recognition tasks. Also, it is likely that the mechanisms used in different top-down tasks are different. There may not be a single unified theory of top-down attention to accommodate all these tasks.

The mechanisms involved in selective attention still remain open in the field of perception research. New discoveries indicate that many areas of the brain share the processing of information from different senses and are multi-sensory in nature. Additionally, the two biological processes mentioned earlier do not take place independently, particularly in goal directed search but instead, they work in a complementary way.

2.2.3 Attention models and categorization

A visual attention computational model is an approximation of the HVA system that describes how the attention in humans or a function works. The existing models usually consider the following aspects: (i) a formal problem statement (ii) the algorithmic and mathematical considerations and, (iii) the implementation level. In [38], the authors made a taxonomy of the current existing computational attention models (bottom-up or top-down) as described next.

2.2.3.1 A taxonomy of computational models

This taxonomy provides a link between the biological and computer vision communities' view of attention. The computational branch is the intersection between the biological vision and the computer vision techniques. Four classes of models were recognized. Here, we only discuss one specific class (i.e., the saliency map) as most of the previous works fall into this category. Furthermore, all the work in this thesis belongs to this category.

2.2.3.1.1 The saliency map

By far the most well-explored area of the HVA and most of the computational models fall into this category. The models from this class are either pure bottom-up, pure top-down or the combination of both saliencies.

However, as mentioned earlier, the bottom-up models dominate the literature, while very few models exist for the top-down approach.

The saliency hypothesis arose from the famous integration theory of Treisman and Gelade [41] and was first described algorithmically by Koch and Ullman [42]. The basis of these models are the following five elements:

1. An early representation of stimuli in the form of features.

2. A topographic map called the saliency map which encodes the combined details from these extracted features.

3. A normalization operation performed on a saliency map to account for different modalities across the features (i.e., different range of values in the features).

4. A winner-Take-All (WTA) network implementation based on the conspicuity, similarity, and domination of the detected salient region.

5. An inhibition of return (IOR) mechanism which ensures progressive shifting of attention to the next best location and prevents fixation on the most salient region.

In addition, there are various other details apart from these five basic elements such as the centre-surround mechanism, feature maps, conspicuity maps which are discussed in section 2.3.1.2.1.

Many computational models follow the Koch and Ullman's algorithmic model with different variations [1, 43–48]. The most noticeable amongst all with the highest number of citations in this research domain is the work by Itti et al. [1]. From now on, we refer to this model as the *Itti* model of attention. More about *Itti* model will be covered in section 2.3.1.2.1.

2.2.4 Object detection

Object detection is one of the essential tasks in image processing and computer vision. In the context of computer vision, the term *object* refers to an entity having attributes which makes it distinct from the background [49]. This distinction comes from pre-existing knowledge of such entities (e.g, boundary, uniformity, shape, etc.). Depending on the application, the definition of object detection varies. If the goal is to determine all the instances of generic objects and their locations, then the detection becomes *generic object detection or class-generic detection*. More recently, such detection involves a measure (called objectness) of how likely an image window contains an object of any class. This kind of detection involves the utilization of object appearance cues such as edges, pixel straddling and location priors [50].

The classic definition of object detection, which is followed throughout the thesis is to find all instances in an image belonging to a particular class, such as pedestrians, faces or planes. Hence, the identity of the object is defined by its class, which in turn depends on the hierarchy of discriminative classes being employed [49]. Typically a small number of object instances is present in an image. In fact, in most of cases, only one is present. This is typically the case in many classification and recognition datasets such as Caltech-256 [51] and PASCAL VOC 07 [52]. Note that alternatively, a slightly modified term for object detection that will be used in rest of the thesis is *target object detection* where the term *target* corresponds to the object of interest to be searched for in an image.

Object detection is the first step that allows gathering further information about the object itself or other content of the scene. It has been used in many applications such as human-computer interaction [53], tracking [54], image retrieval [55] and autonomous vehicles [56]. The scope and challenges associated with object detection are, in turn, defined by the particular applications' requirements.

In previous decades, an immense amount of research has been dedicated to addressing challenging issues in object detection. A comprehensive review of such developments in object recognition can be found in some of the survey literature [49]. A good object detection technique should be robust to viewpoint variation, occlusion, scale, clutterness, and deformation.

Two broad categories of models exist for object detection, *generative* and *discriminative* models. Generative models follow a probabilistic approach where the detection is usually conditioned over the appearance, pose or the visual attributes of the object [57]. The posterior probabilities characterize the likelihood of a region containing a target object. No information about the non-object class (usually the background) is required in this kind of modelling.

In discriminative models, a classifier is learned that can discriminate between the object of interest and non-object regions. The tuning parameters of the classifier are chosen to maximize the detection accuracy [57]. Both models are machine learning based and require some model training to find the optimized parameters for the detection.

2.2.4.1 Research in object detection

In this section, a critical review of passive approaches to the object detection is given here. A common feature of these techniques is that they lack purposeful control over the data acquisition process (hence being passive).

2.2.4.1.1 Deformable or part based detection

This approach deals with the question of whether the humans recognize objects by initially detecting sub-parts of it or the object as a whole in a single shot [58]. In part based methods, localization of important parts of the object play an important role in detecting the whole object. Part based models have proven to be effective in scenarios with partially occluded objects [59]. The limitation of these techniques is the difficulty in extracting and learning part representations from 2D images. In addition, the process is time consuming, particularly for objects that appear in small scales.

2.2.4.1.2 Appearance based detection

Early works in appearance-based detection used global low-level image descriptors based on colour and texture histograms to describe a variety of object poses. However, such approaches were sensitive to illumination conditions and did not perform well in non-ideal imaging conditions. With the introduction of principal component analysis (PCA), a much more compact and robust representation of the varying appearances was achieved [60].

2.2.4.1.3 Local feature-based detection

This is the most popular approach that gained popularity in image classification problems. This is due to its robustness cluttered backgrounds, occlusions and viewpoint variation. In these approaches, local features are extracted at a dense grid from different image locations. This is followed by building a descriptor from these features that are used by a classifier. Scale-invariant feature transform (SIFT) features is an example of local features in which various interest points based on local orientation are extracted from an image using Difference-of-Gaussian (DoG) like operators, and further transformed in a descriptor [61]. Other example features include the histogram of gradient (HOG) features [62], speeded up robust features (SURF) [63] and Harris corner detectors [64].

Another major contribution in the local feature approach was the introduction of the concept of ''bag-of-word'' by Sivic and Zisserman [65] which was widely adopted by the research community in computer vision. The process initiates by extracting local features from different patches of the image using some descriptor (e.g., SIFT, Harris, etc.). The feature vectors are converted into codewords which are feature representation of similar patches. This is typically performed using a clustering technique. Finally, a histogram of codewords generates the final representation of the image or region of the image. The bag-of-word approach is very effective in detecting a single object but not robust when multiple instances of a target is present.

2.2.4.2 Sampling strategies in object detection

A key element in object detection is where to select a region from an image for detecting the object of interest. An ill structured sampling strategy may result in detection accuracy degradation and increase in processing time. The following strategies are adopted.

2.2.4.2.1 Contextual priming

Extensive works exist on how to exploit contextual knowledge of the scene to sample relevant regions of the image [66–68]. For example, in order to detect the sun, it would be more appropriate to search for in the sky rather than on the ground. Contextual priming reduces the number of regions to be processed for further detecting the target object.

2.2.4.2.2 Sliding window

This particular sampling strategy is considered by most object detection systems that use local features and bag-of-words approach [69]. An exhaustive search is applied throughout the image at different scales and locations. The most critical part of this strategy is the classifier which determines the likelihood that a given patch at a [1]particular scale contains the object. The classifier is comprehensively trained over all possible patch scales and sizes. Although this approach is effective when used iteratively for detecting the object, it is computationally demanding.

2.2.4.2.3 Attention based sampling

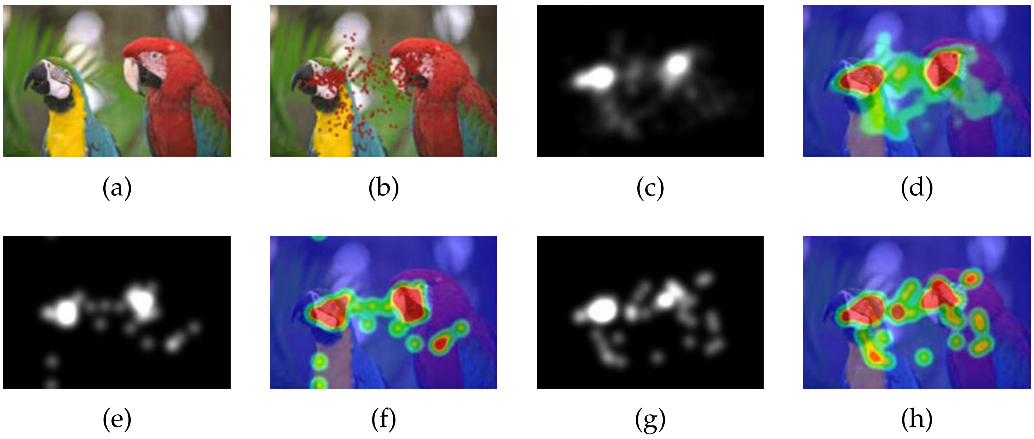
This strategy is based on sampling an image through attending to regions of the image that deem most salient [70, 71]. This approach is very efficient as the process involved in generating feature maps is typically very fast. Furthermore, this approach tends to effectively capture regions of the image that are likely to contain the object in two different modes, a bottom-up and a top-down saliency modes. In the latter, those objects that capture the attention are sampled (not necessarily the object of interest). When high-level object knowledge is imposed, the sampling is conducted through a top-down mode that more likely to sample regions with higher probability than that sampled during the bottom-up mode [6]. This area of research is relatively new, particularly for object detection and constitutes the main goal of this thesis.

2.2.4.3 Performance evaluation

In machine learning, standard evaluation measures and procedure exit in order to evaluate the performance of an algorithm [72]. In this section, we highlight some of these evaluation criteria but in the context of saliency for object detection. Note that in attention based detection, once a saliency map is generated (the outcome in machine learning term), its is compared with the groundtruth map (labelled data). The comparison is done in two dimensions as both the outcome and the labelled data are images (i.e., maps).

2.2.4.3.1 Kullback-Leibler (K-L) divergence and correlation coefficient

To compare two saliency maps, two common similarity measures exist. The first using the Pearson correlation coefficient with the maximum coefficient of one indicating a perfect match and zero indicating lowest similarity. The second uses the unbounded K-L divergence where a value of zero indicates a perfect match. For instance, in Fig. 2.1, the first two images (a,b)



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 2.1: Comparison of two saliency maps: (a) The original image (b) Groundtruth as fixation points, (c) Groundtruth as continuous saliency map (d) Heatmap saliency groundtruth overlaid on top of the original image (e,g) Saliency maps generated through two different techniques (f,h) The corresponding heatmaps respectively.

show the sample image and the groundtruth fixation point overplayed on it respectively. It is a common practice to convert the discrete points into a continuous groundtruth map by convolving these points with a 2D Gaussian filter [73]. The resultant continuous fixation groundtruth map from this operation is shown in Fig. 2.1(c). For visual purposes, this map is represented by a heat map overlaid on the original image and (see Fig. 2.1(d)).

The figure shows two saliency maps generated by two different saliency techniques (see (e) and (g)). When the K-L divergence is computed between the groundtruth map (i.e., Fig. 2.1(c)) and the two maps separately, the obtained divergence values are 3.33 and 7.06 respectively. This shows that the map in (e) is better in predicting fixation than the map in (g) when the reference groundtruth map is (c).

2.2.4.3.2 Receiver operating characteristic (ROC) and precision-recall curves

Another performance measure commonly used in the saliency literature is the receiver operating characteristic (ROC) analysis. In this method [74],

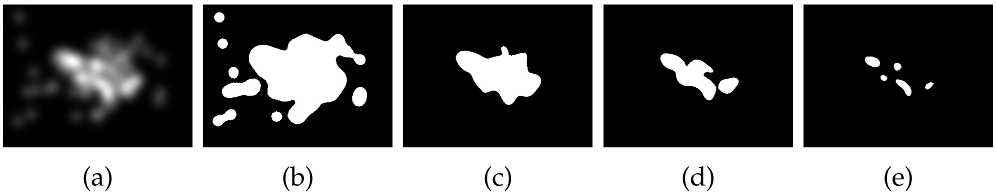


Figure 2.2: Fixed threshold segmentation of a saliency map (a) with a threshold percentage of (b) 90, (c) 60, (d) 40, and (e) 20 respectively.

the groundtruth saliency map is thresholded using a fixed threshold value to convert it into a binary map M_1 (if it is not already in that form). Higher the threshold value, the more pixels from the saliency map having higher intensities are retained as shown in Fig. 2.2. The same thresholding is performed on the saliency map generated by the predicting technique. However, this threshold is different in the sense that threshold values are systematically moved between the minimum and the maximum values of the map and binary maps M_q are generated where $q = 1, \ldots, U$ such that U is the number of thresholded maps. From a pair of M_1 and M_q, the following four parameters are found: true positives (TP), the false positives (FP), the false negatives (FN), and the true negatives (TN).

Once the above mentioned four parameters are acquired, two different plots are generated. The first one is called the receiver operating characteristic (ROC) which is the FP rate (FPR) as a function of the TP rate (TPR) or recall. The second plot is the precision-recall curve. These entities are given as:

$$\begin{gathered} Recall = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \\ Precision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \\ FPR = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \end{gathered} \tag{2.1}$$

Typically for ROC curves, the area under ROC (AUC) is calculated indicating the overall performance of the saliency map. Higher the AUC value, better would be the performance. Similarly for the precision-recall curve, the higher the curve, and in particular, for low values of recall, the better would be the saliency map in identifying the correct saliency locations with respect to the groundtruth map.

Figure 2.3 shows two sample saliency maps along with the groundtruth map (see (b-d) in the figure). The precision-recall and ROC curves are plotted for these two maps using different threshold values. The blue curves that correspond to the saliency map shown in (c) indicate a better performance than those associated with the feature map in (b) in terms of detecting the salient region.

In order to evaluate the performance of a particular technique for saliency detection, these curves are plotted across all the images in a dataset. For each threshold value, the computed values in averaged over all the saliency maps, resulting in much smoother curves than that plotted for a single saliency map (see Fig. 2.3(g,h) that are plotted over the entire dataset).

2.2.4.3.3 The *F*-score

Another common performance measure is the *F*-measure which is a function of precision and recall. It can be calculated over a set of threshold values as in AUC-ROC or precision-recall curves, or over a single threshold value. A single threshold value is usually selected adaptively depending on the input image for which the saliency is calculated. Higher *F*-measure values correspond to better performance. The *F*-measure is given as:

$$F_\beta = \left(1 + \beta^2\right) \cdot \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall} \tag{2.2}$$

where β is an importance factor for weighting either precision or recall. A common value for β^2 is 0.3 [74] in the saliency literature.

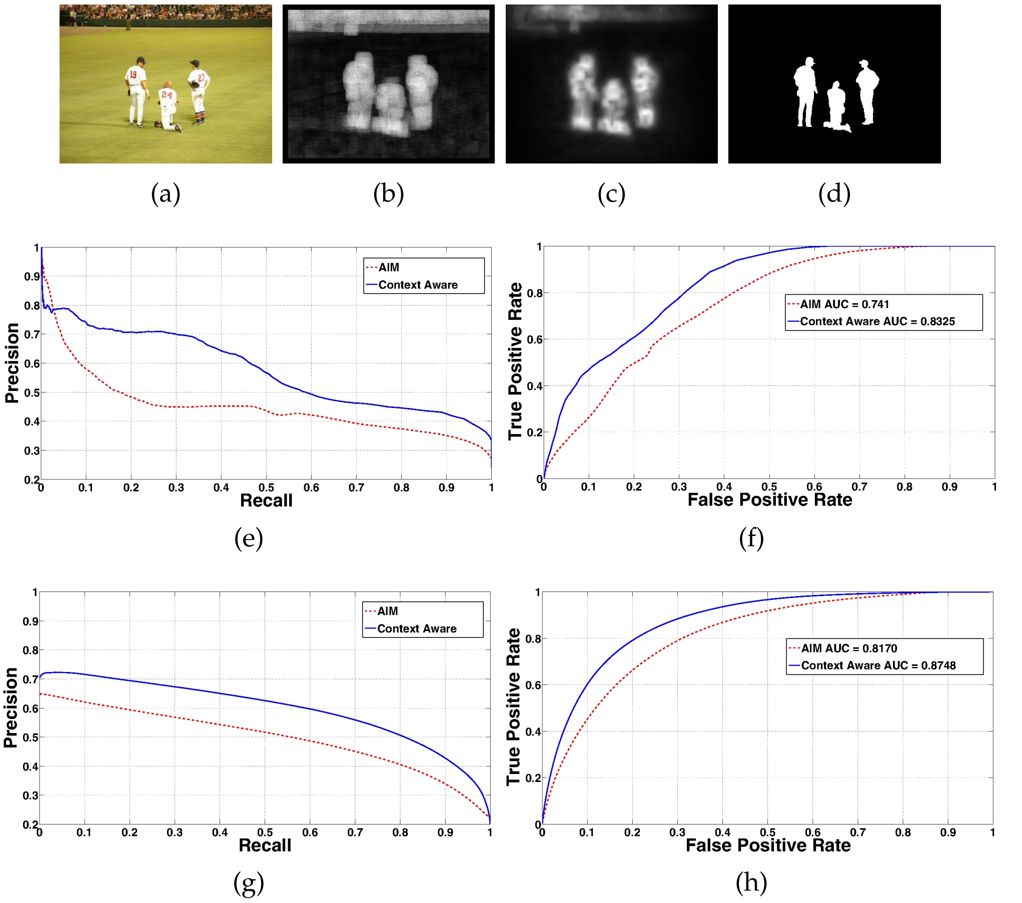


Figure 2.3: Precision-recall and ROC performance measure curves, (a) Input image (b) Saliency map by AIM [75] (c) Saliency map by Context Aware [76] (d) Groundtruth (e) Performance measure of both the techniques on the original image through precision-recall curve and (f) ROC curve (g) Precision-recall curve for both techniques on the ASD dataset and same for (h) ROC curve.

There are many ways to select the adaptive threshold value. The most common one is based on the method proposed by Achanta et al. [74]. Initially a saliency map is segmented using mean-shift segmentation with the help of the original image. The segmentation technique has three parameters, σ_S, σ_R and σ_{min} which are set to the values 7, 10 and 20 respectively as chosen in [74]. Once the map is segmented, it is thresholded using an adaptive threshold value T_a given as:

$$T_a = \frac{2}{W \times H} \sum_{x=0}^{W-1} \sum_{y=0}^{H-1} S(x, y) \tag{2.3}$$

where W and H are the width and height of the saliency map respectively and $S(x, y)$ is pixel intensity of the saliency at position (x, y). This threshold value is used to binarize the saliency map.

There are many other measures for saliency performance evaluation, however, in this chapter, we include only those that will be used throughout the thesis for evaluating the performance of the proposed models. For a comprehensive survey of saliency evaluation techniques, refer to the work by LeMour et al. [77].

2.3 Related work

This section provides a detailed information about bottom-up and top-down saliency and previous computational models of both processes.

2.3.1 Bottom-up saliency

The purpose of the bottom-up saliency models is to generate bottom-up maps that either highlight the most salient regions in a scene or to predict where human fixate while free viewing the image. The maps generated from the former category are typically called saliency maps and fixation maps from the latter. Although the research work in this thesis is concerned with modelling top-down saliency, it is essential to understand the working of bottom-up saliency as it will assist in perceiving the significance of top-down saliency.

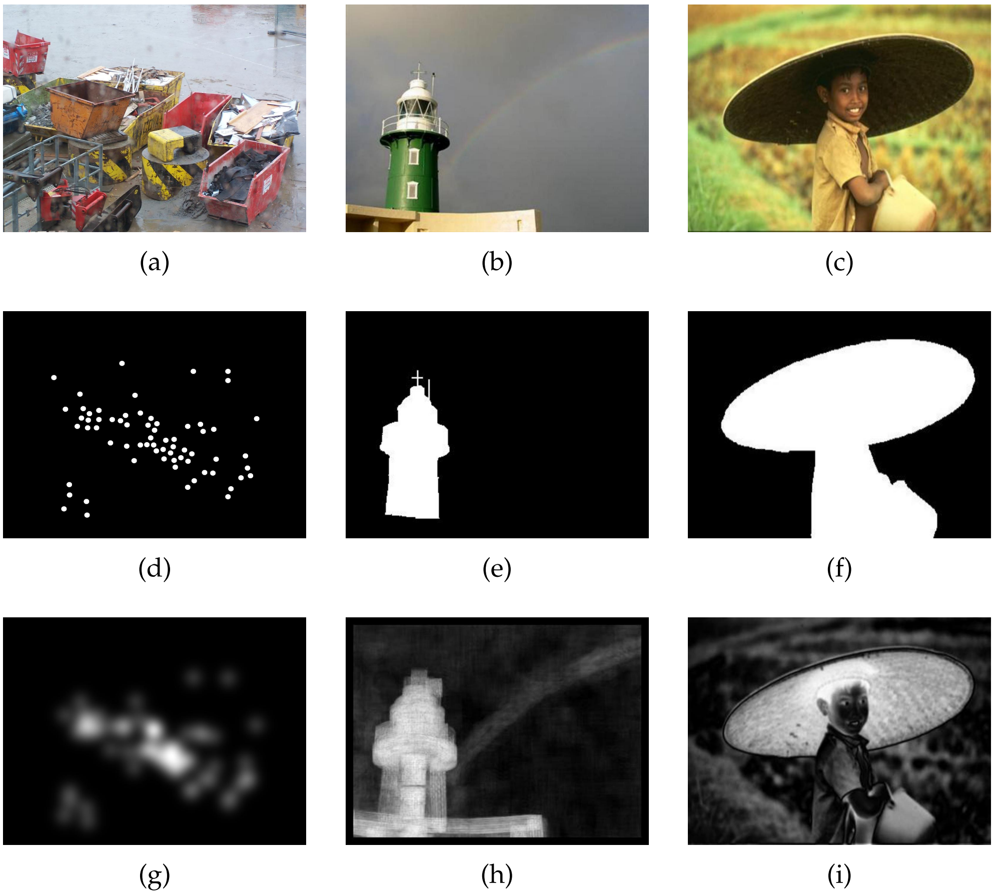
2.3.1.1 Saliency and human fixation maps

This section briefly explains the visual difference between the two types of maps through an example. The three sample images in Fig. 2.4 (see the first row) are selected from two datasets. The first image (i.e., Fig. 2.4(a)) is chosen from a fixation dataset while the other two are drawn from a salient object detection dataset. In addition, the second row of the image represent the corresponding groundtruth maps of the sample images for reference. The final row shows the corresponding saliency or fixation maps generated through the following techniques, from left to right, the fixation prediction technique proposed by Judd and Torabla [73], the fixation prediction technique proposed by Bruce and Tsotsos [75], and the salient object detection technique by Achanta [74].

The map produced by the object detection technique (see Fig. 2.4(i)) is smooth, compact and highlights the salient objects as a whole. The intensity of a point on the map reflects its saliency level. Furthermore, the groundtruth maps for those datasets used to evaluate the performance of salient object detection techniques represent segmented regions (e.g., Fig. 2.4(f)).

The fixation maps as shown in Fig. 2.4(g) tends to be blurry and scattered and mostly used to predict human fixation in a free viewing task. The groundtruth fixation maps (see Fig. 2.4(d)) represent discrete fixation points that are gathered from subjects who were asked to view images freely. Note that concentrations of fixation points are most likely within salient regions.

A fixation prediction technique such as the one proposed by Bruce and Tsotsos [75] can also serve the purpose of salient object detection (see Fig. 2.4(h)). Most fixation techniques can perform well in detecting salient objects because humans tend to fixate at the most salient region of the



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 2.4: Saliency and fixation maps: (a) –c) Original images (d) –(f) Groundtruths, and the saliency maps generated by the following techniques (g) Judd and Torabla [73] (h) AIM [75] (i) Achanta [74].

image when free viewing. However, even then, the visual appearance is blurry compared to those produced by salient object detection techniques.

2.3.1.1.1 Saliency datasets

There exist various saliency and fixation datasets for performance evaluation. The complexity of the datasets varies in terms of background clutter, multiple saliency objects, the size of objects, position of the salient regions and blurriness level of the salient objects. The saliency datasets are used explicitly for evaluating the performance of bottom-up saliency techniques, where the objective is to detect salient objects.

Most of these datasets have some shortcoming. For instance, the salient objects are typically located at the centre of the image. These salient objects are well photographed and focused. In addition, they lack saliency competitiveness which means that in these datasets, there is often a lack of non-salient distractors sharing similar features to the salient objects. Table 5.1 summarizes the most popular datasets for both saliency and fixation prediction.

2.3.1.2 Biological inspired bottom-up techniques

Many biological plausible bottom-up saliency models and techniques have been proposed in the past. Almost all these models are directly or indirectly inspired by the cognitive concepts. One of the leading works in biologically inspired bottom-up saliency is by Itti et al. This model has been the basis of later models and the standard benchmark for comparison. Some of the proposed top-down saliency models in this thesis are based on the general structure of *Itti* model.

2.3.1.2.1 The *Itti* model of attention

As shown in Fig. 2.5 [1], the model takes the input image and creates nine scales using dyadic Gaussian pyramids [84]. Sub-sampled images are pro-

Table 2.1: Saliency and fixation datasets

Name	Type	Groundtruth	Size	Description	Reference
Benchmark	Fixation	N/A	300	These natural images with eye tracking data from 39 observers are used by the fixation community to evaluate the performance of any newly proposed fixation technique. The ground truth is no available publicly. It is considered the benchmark dataset for fixation	Judd et al. [78]
MIT	Fixation	Fixation points and continuous maps	$1,003$	It is purely meant for fixation. Many images in this dataset do not have salient objects. The ground truth data are constructed by collecting eye tracking data of 15 viewers	Judd et al. [73]
Toronto	Fixation	Fixation points and continuous maps	120	The images are viewed by 11 subjects with free-viewing task. The images are both from outdoor and indoor scenes. A large portion of images do not contain particular regions of interest or salient regions.	Bruce and Tsotsos [75]
MSRA	Saliency	Rectangle boxes	$20,000$ and $5,000$	This is a huge dataset containing two sets, a smaller containing $5,000$ images and the larger with $20,000$ images. The labeling is done by nine and three subjects respectively	Tie Liu et al. [79]
ASD/MSRA-1000	Saliency	Segmented objects (Binary maps)	$1,000$	This is the most popular dataset for saliency. The images in this dataset contain a single salient object per image.	Achanta et al. [74]
ImagSal	Saliency	Segmented objects (Binary maps)	235	The 235 images are divided into six different categories. These categories contain images with large, medium and small objects. Also some images contain repeating distractors, multiple salient objects, cluttered background and salient regions with different sizes	Jian Li et al. [80]
IRCCyN IVC Berkeley Eyetracker	Saliency and fixation	Pixel-wise ground truth, fixation points and importance maps	80	These images are collected from the Berkeley segmentation database. 80 of these images were selected and 24 observers viewed these images and an eyetracking device was used to collect the eye tracking data.	Wang et al. [81]
SOD	Saliency	Pixel-wise segmentation	300	This dataset is a collection of salient object boundaries based on Berkeley Segmentation Dataset (BSD). Seven objects are asked to choose the salient object(s) in each image. Each subject is shown randomly a subset of the Berkeley segmentation dataset as boundaries overlapped on the corresponding images.	Movahedi and Elder [82]
SED	Saliency	Pixel-wise ground truth	200	This dataset contains two sets of images each of 100 images. The first one contains images having a single salient object while the other set contains images with two salient objects.	Alpert et al. [83]

duced by performing a progressive low pass filtering on these images.

This is followed by extracting three types of features (we call them channels) using various filters. These are colour, intensity and orientation features. The features are computed through biologically inspired phenomena known as the centre-surround mechanism akin to the visual receptive fields found in the visual cortex (refer to section 2.2.2.1 for the On-OFF neuronal response in the retina region). The centre-surround is implemented as the difference between fine and coarse scale features. The centre pixels are always chosen at the scales belonging to $2, 3, 4$ and the surround at $5, 6, 7, 8$.

The difference between two features is performed across-scales, yielding a multi-scale feature extraction. The features are calculated with varying parameters within a channel. For colour features, two contrast features are calculated, the red/green and blue/yellow. Only one feature is extracted for intensity. Finally, for orientation, Gabor filters are used to extract the orientations at $0^o, 45^o, 90^o$ and 135^o. The following equations summarize the feature extraction across scales:

$$\begin{aligned}
\mathcal{I}(c,s) &= I(c) \ominus I(s) \quad \text{for intensity}\\
\mathcal{R}/\mathcal{G}(c,s) &= |(R(c) - G(c)) \ominus (G(s) - R(s))| \quad \text{for R/G contrast}\\
\mathcal{B}/\mathcal{Y}(c,s) &= |(B(c) - Y(c)) \ominus (Y(s) - B(s))| \quad \text{for B/Y contrast}\\
\mathcal{O}(c,s,\theta) &= O(c,\theta) \ominus O(s,\theta) \quad \text{for orientation}
\end{aligned} \tag{2.4}$$

where c and s represent the centre and surround scales whereas θ is the angle used in computing the local orientation using Gabor pyramids. The operator $\ominus$ represents the across scale difference. In total, we have 42 maps (referred to as feature maps (FM)) produced from the multiscale feature extraction process, six for intensity, 12 for colour and 24 for orientation.

The obtained feature maps consist of values that differ from one category to another and their modalities become incomparable. As a result, Itti et al. proposed a normalization operation that globally promotes maps having unique or strong peaks. The operator indicated as $\mathcal{N}(.)$ finds the

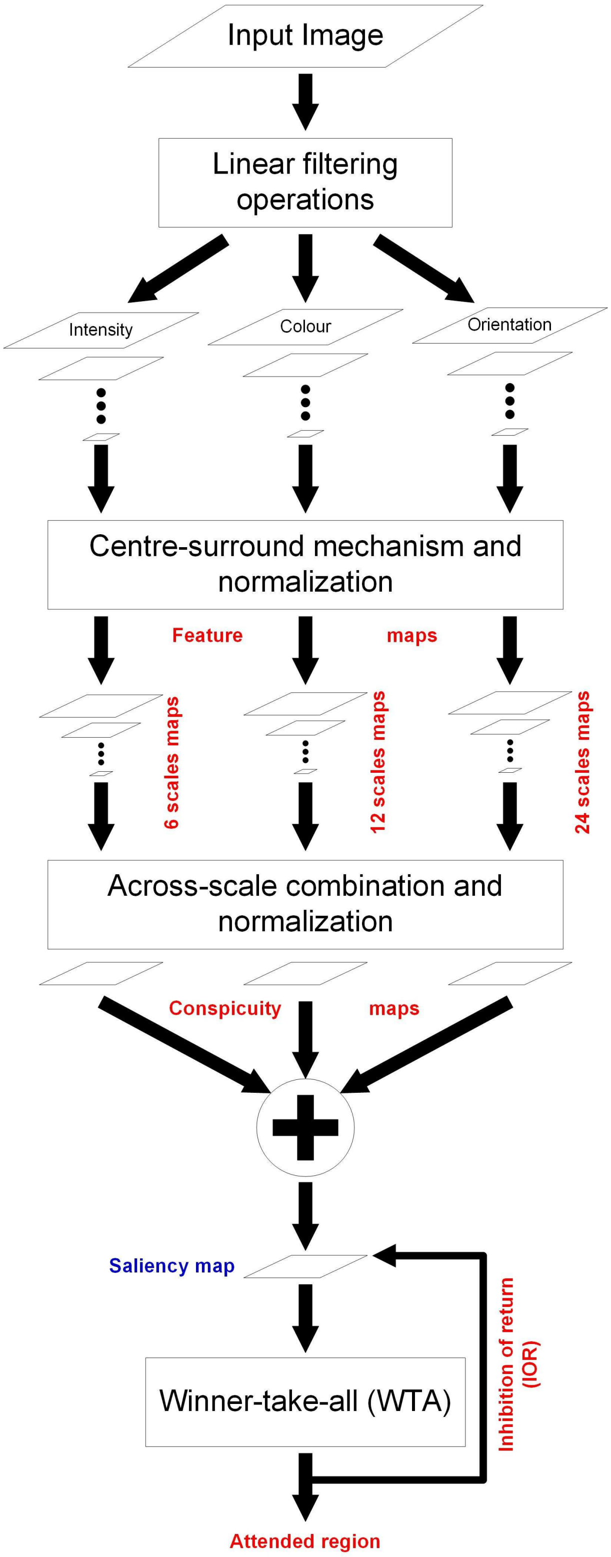


Figure 2.5: *Itti* attention model for fixation. This diagram is a redrawing of the model presented in [1].

global maximum M in a map and computes the average $\bar{m}$ at other local maxima, and then globally multiplies the map by $(M - \bar{m})^2$.

The feature maps are combined through across scale addition operation ($\oplus$) to yield the conspicuity maps (CM) as follows:

$$\begin{aligned}
\bar{\mathcal{I}} &= \bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{c+4} \mathcal{N}\big(\mathcal{I}(c,s)\big) \quad \text{for intensity CM} \\
\bar{\mathcal{C}} &= \bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{c+4} \big[\mathcal{N}\big(\mathcal{R}/\mathcal{G}(c,s)\big) + \mathcal{N}\big(\mathcal{B}/\mathcal{Y}(c,s)\big)\big] \quad \text{for colour CM} \qquad (2.5) \\
\bar{\mathcal{O}} &= \sum_{\theta \in (0^o, 45^o, 90^o, 135^o)} \mathcal{N}\Big(\bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{c+4} \mathcal{N}\big(\mathcal{O}(c,s,\theta)\big)\Big) \quad \text{for ori. CM}
\end{aligned}$$

The reason for having different conspicuity maps each normalized independently because similar features compete strongly while different features vary in their contribution toward building the final saliency map [1].

Finally, the saliency map is generated by combining the three maps using any operator. The default is through summation but various other versions exist and have been followed in other models that are based on the *Itti* model. The saliency map generation is followed by winner-take-all(WTA) and inhibition-of-return (IOR) mechanisms for fixation and gaze shift respectively.

WTA network is a computational principle in neural network models by which neurons compete with each other for activation. In the context of saliency maps, those areas having higher activation (i.e., intensity values) stays active while all others shut down. The focus of attention (FOA) is shifted to the location of the winner neuron. The WTA is complemented by another process called the inhibition of return (IOR). In IOR, suppression of processing of previously inspected locations takes place. This allows the next most salient location to subsequently become the winner. It also prevents FOA from immediately returning to the previously attended location.

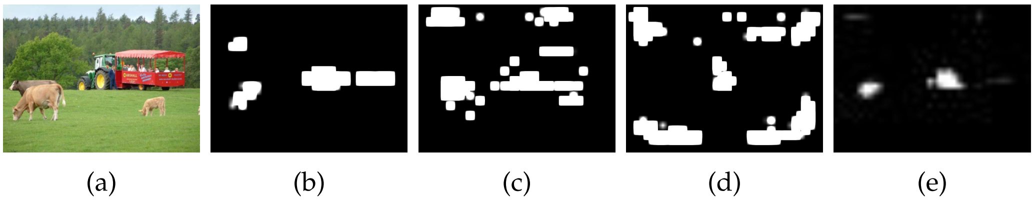


Figure 2.6: The conspicuity and the final saliency maps of *Itti* model, (a) Original image (b) Conspicuity colour map (c) Conspicuity intensity map (d) Conspicuity orientation map (e) Final saliency map.

As an example of the saliency map and the fixation locations from this model, Fig. 2.6 shows the three conspicuity maps and the final saliency map of the image in Fig. 2.6(a). Figure 2.7 shows four fixations at consecutive times modeled by the WTA-IOR network on the image of Fig. 2.6(a). The first row of the figure i.e., (a–d) shows the first four saccade shifts displayed on the original image. The four WTA saliency maps (i.e., maps after suppressing the locations not attended to by IOR) representing the current fixation is shown in the second row of the figure i.e., (e–h).

The third row of Fig. 2.7 shows the most prominent feature map at a particular scale that contributed to the construction of the WTA map for a fixation (see i.e., (i–l)). For instance, for the first WTA map, the Blue/Yellow feature of the colour channel at a scale of $(7-3)$ contributed the most amongst all other features at different scales to yield the WTA map shown in Fig. 2.7(e).

By default, the *Itti* model is entirely data-driven. However, if weights are associated with the features at the feature conspicuity levels, then it can be tuned for a specific task. More about this is covered in section 2.3.2.2.

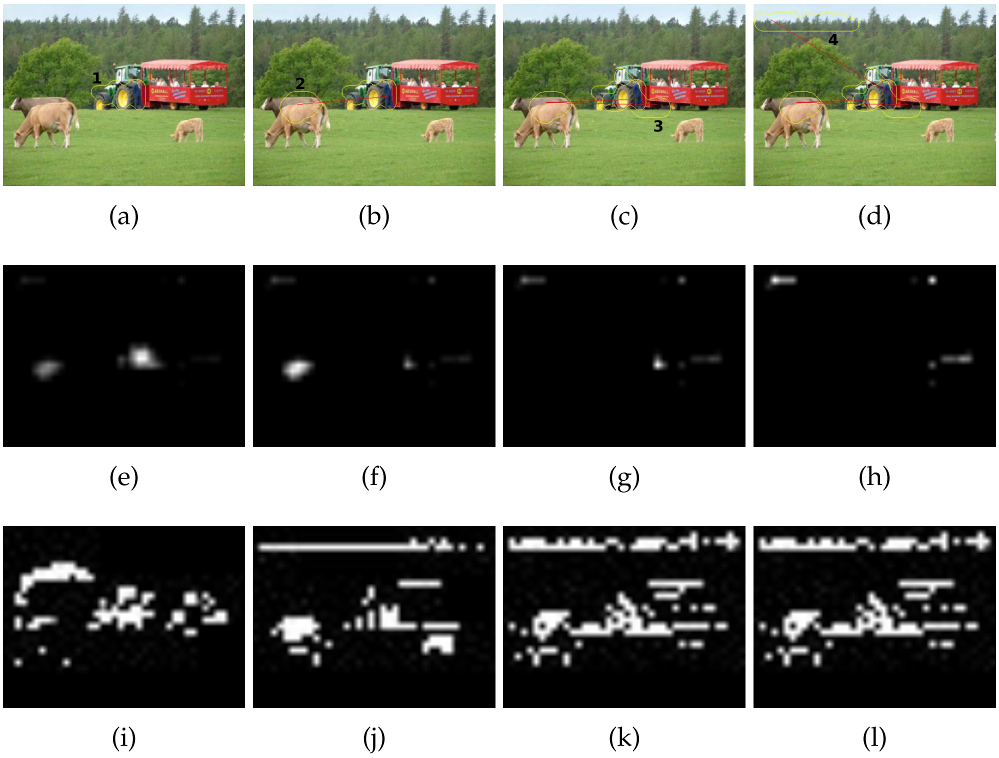
Finally it is worth mentioning that all the terminologies used by Itti et al. to describe various components, processing steps and entities are commonly used in visual attention literature. Since our proposed techniques, specifically in chapters 3-5, are highly dependent on the Itti model, we have adhered to the same terminologies. A slightly different terminology is followed in chapter 6 which is highlighted in section 6.1.1.

Three of the terminologies are extensively used throughout the thesis. *Feature maps* (FM) are referred to 2-D maps that are generated after applying a centre-surround and normalization mechanism at multiple scales on the extracted features. On the other hand, *conspicuity maps* (CM) are also 2-D maps that are generated by combining FMs belonging to similar category features at multiple scales (e.g. Red, green and blue feature maps are combined together to yield a colour conspicuity map). Note that more than one FM exist and at different scales. However, only a single CM exist at one particular scale. When different conspicuity maps are combined followed by a normalization process, a *(*saliency map) (SM) is generated. Hence, SM is a 2-D the final product map that highlights various regions of interest.

2.3.1.3 Other biologically plausible techniques

Inspired by Itti's work, many of the works followed this model both in bottom-up and top-down saliency calculation. In this section, only bottom-up saliencies are discussed. In [43], authors proposed a similar structure model and incorporated other features such as contrast sensitivity functions, perceptual decomposition and visual masking with a similar centre-surround mechanism. They further improved their model by incorporating achromatic, chromatic and temporal information [44]. Bian and Zhang proposed a biologically inspired technique but in the frequency domain [45]. The model was much faster than *Itti* model as the spatial domain was avoided. They used Laplacian pyramids and overlapping local patches to generate the centre-surround effect.

In addition to the conventional features, in [85], authors used the symmetry features for finding salient regions. They developed three symmetry-saliency operators based on the isotropic symmetry, radial symmetry, and colour symmetry. They showed that their model performed significantly better on symmetric stimuli than the method of Itti.



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

Figure 2.7: WTA-IOR network for consecutive time fixations, (a–d) Estimated final fixation regions (e–h) WTA-IOR final maps (i–l) WTA feature map having the most prominent contribution at an instance of time.

Recently sparse coding techniques have been used for feature extraction. In [46], using a set of Independent Component Analysis (ICA) filters, features are extracted by convolving these filters with the input stimuli. The basis functions are learned through the eye-fixation patches from an eye-tracking dataset. Clutter and local contrast features along with ICA features were also used. A similar technique employed sparse coding for feature extraction in aerial images [47].

The main advantage of biologically plausible models is that they accelerate the understanding of computational principles and methods that are found in HVA system for complex processes such as object detection and recognition.

2.3.1.4 Computational bottom-up techniques

Most of the proposed techniques for bottom-up saliency are purely computational in nature. However, many of them have structures similar to the biological ones (i.e., feature extraction and integration based). The computational techniques for saliency are inspired by various engineering and computer science areas, such as but not limited to machine learning, evolutionary computation, signal processing, image analysis and information theory. Some of the previous bottom-up computational saliency techniques are given below.

Starting with techniques which are implemented in frequency domain, Achanta et al. work is probably the most popular [74, 86]. They use low-level colour and luminance features based on the *CIELab* colour space. With the help of the centre-surround mechanism, lower or higher frequencies are retained depending on the size of the centre-surround filter being used. Another fast and simple frequency-based technique was proposed by Hou and Zhang [87]. Their model is independent of features, categories, or other forms of prior knowledge of the objects. A spectral representation was extracted from an image through the use of the log-spectrum. Most of the frequency based saliency techniques are fast. However, one main disadvantage is that such techniques capture the contours and edges of the salient objects only and not the whole salient region.

Another set of techniques uses Bayesian models to use prior knowledge of the scene such as context to find regions of interest. Examples of this class include the work of Jinhua Xu [88] in which the bottom-up saliency is defined as the joint probability of the local feature and context at a location in an image. The Bayesian Surprise theory of Itti and Baldi [89] is another Bayesian model that defines saliency as a deviation from what is expected based on a set of internal local visual features. They measured the surprise element using the KL divergence metric.

Machine learning techniques played a significant role in improving the saliency detection by learning the saliency parameters through different learning procedures [90], [73].

Another important class of computational models uses graphical techniques for saliency extraction. In [91] for instance, the authors used a graph-based technique to perform the normalization of the feature and conspicuity maps acquired from *Itti* model. They show that their normalization method produces better saliency maps than Itti's normalization method.

Global and local region-based techniques have greatly captured the interest of many for saliency detection. In [92], authors proposed a salient detection and segmentation technique that uses local colour contrast histogram along with spatial region histogram for object detection. Other techniques use some metrics to find the similarity between image patches [76, 93, 94]. In [76], patches are extracted from the original images and dissimilarity measure consisting of centre, spatial and colour distances are used to distinguish a salient region from a non-salient one with the help of context within the neighborhood of each patch.

Some models are information theoretic. Information theoretic concepts are closer to interpreting biological approaches for bottom-up saliency. It can be hypothesized that a region or an object is salient for an individual if it contains more useful information than the other regions or objects in an image. This hypothesis has a biological flavor as a salient region could be any region that may differ from an individual's perspective. This difference of choice could be seen as an information variation [6] or as an element of surprise [89].

There are a number of existing approaches on saliency based on information theoretic concepts. In [75], the authors proposed a method based on information maximization. The information is computed based on Shannon's self-information. In [95], a novel approach combining both graph based and information theory is proposed.

One disadvantage of these information theoretic models is that only self-information was considered of a region or feature and no difference of information was considered. To overcome this disadvantage, authors in [96] considered the difference of information between patches using the KL divergence. They called this information difference *information divergence*. Their method is based on extracting features through ICA and then applying the information divergence in a centre-surround mechanism. Their model has achieved good results on both saliency and fixation datasets. A thorough discussion about this model will be presented in chapter 3.

A comprehensive detail about these techniques and many other bottom-up classical saliency techniques are given in [97] and in the MIT saliency benchmark report [78]. In recent years, research in bottom-up saliency has flourished considerably. A review on some of the current state-of-the-art techniques for bottom-up saliency detection can be found in [27].

2.3.2 Top-down saliency

This section gives an outline of the previous work in top-down saliency and the combination strategies with the bottom-up saliency. Furthermore, the section briefly highlights the target object detection task in the context of top-down visual attention.

2.3.2.1 Neurological studies on top-down saliency

Most of the work in top-down influence is in the experiment phase. Various experiments have been conducted to study the behavior of this influence particularly in high-level tasks such as object detection and recognition. The top-down influence is a voluntary task-driven process that is initiated from high-level areas within the visual cortex. However, there is no existing model that describes the exact behavior of this influence adequately. In this section, some of the experimental studies in the past are discussed.

In [30], the authors indicate that the target template and the context of the scene can be utilized by the visual system from the beginning of the scene viewing to effectively search for the target. The more detailed the representation of the target, the more efficient the ensuing search. Different experiments were performed based on placing target objects at different locations with and without consideration of image context. The experimental results through gaze and fixation monitoring supported their hypothesis. In [98], the authors conducted six experiments based on the representation level of the target. They also concluded that information about the identity of the target plays an important role in how fast the top-down influence changes the selection process. Hence templates or top-down knowledge biases the visual attention system towards targets having similar information.

However, in [99], the authors concluded that subjects set up the target in their brain in a more detailed and precise way rather than in a semantic way. Hence, the top-down influence can be modelled either through a precise description of the target or through a semantic representation.

Another set of studies is concerned with the nature of the top-down influence in the presence of distractors. In [100], experimental studies were conducted to analyze the complexity of finding the letter 'N' amongst reversed 'N' distractors and vice versa. A dense array of distractors was used with a single unique target. Observers usually located the target in half a second in both scenarios. In this experiment, the researchers concluded that the bottom-up influence is responsible for discriminating the target amongst the distractors. However, they also indicated that there is a slight delay in getting to the target due to the interference of the top-down knowledge at the time of fixation over the target.

Hence, they concluded that the bottom-up influence is responsible for detecting the target whereas the top-down has a negative impact on the detection. In the second experiment with less dense search array, the clash between top-down and bottom-up was less obvious as the top-down normally dominated the search while bottom-up has a weaker influence. It is obvious from this experimental work that the nature and the density of distractors play an important role not only in finding the target but also in judging the clash between both influences.

In another work [101], the authors studied the effect of distractors on detecting the targets from density, spatial distance, and specificity of the referential task point of view. They concluded a similar hypothesis as the one made by the authors in [100].

In [102], the authors suggested that there are some limitations in these studies. For instance, it is not known whether combining all these information would facilitate search or not, and if it does, then how these sources of extra information are utilized. Through experiments, they concluded that the visual system treats template information and scene context independently. How these two are combined with the bottom-up saliency to facilitate the search for a target remains elusive.

As is evident from the previous paragraph, the interaction between the top-down influences and the bottom-up influences remains one of the most controversial topics in attention. Several researchers have made an effort to explain the nature of such interaction. In [103], the authors demonstrated that the top-down influence in a task-driven search can rapidly override the bottom-up saliency i.e., within first few fixation, and the role of the bottom-up influence after that diminishes.

In [40], the authors hypothesized that the top-down influence no matter what the task is, cannot override the attentional capture through the bottom-up influence. They emphasized that at every stage in the search mechanism, both influences interact with each other actively, and the influence of the bottom-up cannot be neglected at any point.

The nature of attentional selection in scenes is a dilemma, and researchers have different opinions. For instance, in [104], authors argued that the subjects when viewing a scene tend to fixate on the centre of the objects. They provided evidence that saliency does not drive the attention directly, but it is always in association with the objects. Hence, it is not the saliency which is the driving force but rather the objects. Einhauser et al. [105] also explored an alternative hypothesis in which they suggested that the bottom-up saliency maps based on early features such as orientation, colour, and contrast do not drive the attention directly particularly in high-level tasks such as object detection. The observers instead attend to interesting object, suggesting that the objects themselves predict fixations and not the saliency.

To summarize the direction of the past studies on top-down influence, the majority of the work is experimentally based with many contradictory views. The main questions that the authors addressed are as follows:

1. How the top-down influence is described in term of the guidance sources (e.g., context, template, saliency, etc.).

2. How low-level signals are modulated under the influence of a task.

3. How the guided search for a target is affected by the presence of various kinds of distractors.

4. What role does the bottom-up influence play in the visual search for a target and in recognition.

5. Finally how both bottom-up and top-down saliencies interact with each other and what is the nature of this interaction.

Based on these experimental studies, several attempts to implement various approaches and models were seen in the past. In the next sections, we explore these various approaches in detail.

2.3.2.2 Weighting of bottom-up features

Several computational models exist that describe the top-down influence through weighting the bottom-up features. These model are influenced by the biological behavior within the visual cortex, where certain signals have a modulatory effect on other signals during a high-level task such as object detection.

Probably the most prominent model in this category is the Visual Object Detection with CompUtational attention System (VOCUS) proposed by Frintop [21,106,107]. The model follows *Itti's* the approach in extracting bottom-up features. However, some computational differences include the conversion of the colour space into Lab and normalizing the features maps through dividing the maps with m such that m is the total number of maxima that exceeds some threshold value. This is done to ensure that the detection with highest values pop-out as the most important region of the image. Once the bottom-up saliency is computed by combining the conspicuity maps, the top-down weights are learned in the learning phase.

In the learning mode, regions-of-interest (ROI) from various training images are extracted either through a classifier or manually. These ROIs contain the target object to be searched for in the search phase. This is followed by the computation of the bottom-up saliency again of the training images and the most salient region (MSR) within the ROI. The weights of individual features and conspicuity maps are calculated by taking the ratio of the average saliency of the MSR and the saliency of rest of the regions (background) for that feature or conspicuity map. The weights are averaged over some training images (i.e., those having best detection performance and the worst detection performance).

Once the weights are learned, and before finding the top-down saliency map, these weights are further categorized into two classes, those which belong to the excitation map (i.e., weights with values greater than one representing effective features) and those belong to inhibition map (i.e., weights with values less than one representing inhibitory features). Hence, these two maps are found by applying the weights to the bottom-up features separately to get the excitation and inhibition maps. The final top-down saliency is the absolute difference of the maps.

There are two shortcomings of this model. First, the weights cannot be generalized for different images with various backgrounds and targets. Secondly, there is no context information being incorporated into the system. Hence, in the learning phase, one single set of weights is found for the entire dataset.

Another popular weighting model was proposed by Navalpakkam and Itti [22, 108]. This model considers the knowledge of the distractor along with the target to calculate the weights. For bottom-up saliency features, they used the *Itti* model. The weights are further optimized to get the top-down saliency. The optimization problem is constructed as maximizing the signal to noise ratio (SNR) such that the signal represents expected saliency energy of the target and the expected saliency energy of the distractors is regarded as noise. The weights are calculated within feature dimension indicated by $g_{i,j}$ (i.e., sub-feature j of a feature channel i) and across features g_j (i.e., conspicuity maps) where j is the conspicuity map. The optimized values of these weights are given as [22]:

$$g_{i,j} = \frac{\mathrm{SNR}_{ij}}{\frac{1}{n}\sum_{k=1}^{n}\mathrm{SNR}_{kj}} \tag{2.6}$$

$$g_j = \frac{\mathrm{SNR}_j}{\frac{1}{N}\sum_{k=1}^{N}\mathrm{SNR}_k} \tag{2.7}$$

where n is the number of sub-features within a feature channel and N is the number of conspicuity maps. These weights are calculated in the training phase by averaging the weights over 50 training images. Hence, a feature is relevant and receives a high weight if it renders the target more salient than the distractors, and irrelevant otherwise.

Although this model has good performance when tested on 750 different artificial and natural images, it has some drawbacks. First, as in case of VOCUS, the weights cannot be generalized particularly because both training and testing of the proposed model were done on similar scenario and background images. For instance, in both training and testing phases, the distractors and targets were set in a grid of 9×9 by varying the position of the target and distractors with slight change of background orientation, jitter, and colour. Hence, the overall background and context of the scene were static. Furthermore, this model does not make a distinction between the inhibition and excitation weights as in the VOCUS model.

In another approach [109], the authors proposed a saliency measure called the Composite Saliency Indicator (CSI). This indicator judges the worth of an area within the generated saliency maps for a particular feature map to be a true candidate salient region. In other words, it is used to weight the feature maps according to the quality of the saliency map. Two parameters are used to measure the quality: spatial compactness and saliency density. The compactness is measured through the spatial relation of the salient points in a map and the pixel values are used to measure the density. Based on these measures, related and unrelated maps are separated based on some threshold value.

Only the related maps are considered when generating the final saliency map. Furthermore, weights based on the two previously calculated measures are assigned to each related map for a final linear combination strategy to generate the saliency map.

This approach is a dynamic mechanism for eliminating unimportant features. However, there are two main disadvantages of this approach.

First, although this approach is dynamic, the dynamicity of weights are not learned, but rather evaluated from the current saliency maps. Hence, the calculated weights are generated based on local feature maps and no other information. Secondly, the compactness and density may not truly measure the quality of the saliency map alone. Other measures such as shape, size and edges profile can play a significant role in measuring the quality of a saliency map.

A recent work in weighting is proposed by Benicasa et al. [24]. This work uses the basic features of *Itti* model along with some mid-level features such as size, recognition indicator, and location. In addition, the weights are not learned but adjusted manually to get the desired result. The proposed work follow the object based saliency models [104,105]. The authors claim that the model could be generalized to detect target objects as well. Furthermore, the recognition indicator is used to indicate the likelihood of a segmented region belonging to an object which is deemed to be salient.

This recognition indicator is generated from a high-level classifier tuned to detect different objects. Furthermore, the model is heavily dependent on a segmentation technique as a preprocessing step. Finally, the authors used a similar excitation and inhibition operations to create maps as in the VOCUS model. The major drawback of this model is that it lacks quantitative performance analysis. Furthermore, the weights are not adjusted manually and there are many tuning parameters need to be adjusted before generating the final saliency map. In addition, the modified weights belong to the conspicuity maps and not the sub-features.

In the work produced by Palomino et al. [110], the authors proposed a model for a practical application for finding balls in mobile robots. The model extracts various low and high-level features including shape features, proximity features and target specific colour features and combine them to generate the final saliency map. The model does not follow the multi-scale feature combination as strategy opted by Itti, instead the feature are extracted, weighted and then directly combined. The weights of the features are tuned manually for detecting the target balls. The model lacks automation as the weights are not learned.

2.3.2.3 Combining bottom-up and top-down saliencies

Some models exist that combine both the saliencies to maximize the detection of the target. There is no clear evidence from the theoretical point of view on how the combination takes place and how the combination process benefits the visual search and the recognition task in particular. For instance, if the purpose is to have a vision mechanism only to predict the target in scene, then no attention based visual processing is needed (i.e., no saccadic movement or shift of attention). This requirement is very much the norm in most bottom-up saliency techniques. In this situation, it would be a simple matter of combining the two saliencies linearly or using some predefined combination strategy. However, and keeping in mind that bottom-up strategy is meant for detecting salient regions while top-down for more specific high-level target detection, it remains a controversial question that how would bottom-up saliency benefit the top-down influence as it seems that only top-down is enough for object recognition without the need of bottom-up.

For an active visual search system (e.g., an active vision-based robot) capable of performing saccade movement, fixation and object search, an attention based processing for understanding the scene and further interacting with it is needed. Furthermore, in this case the recognition task is accompanied by gaze shift from one location to another depending on the current objective. For instance, if the purpose is to search for a ball but the at the same time look for other interesting objects, then the bottom-up saliency would actively participate in the combination process.

If the objective is to find the target without attending to other regions, then the bottom-up would less obviously be needed to participate in the combination process. Hence, how and when the saliencies are combined depends on the requirement.

In [106], Frintrop et al. applied a simple time gain on both the maps by linearly combining them to get the final saliency map. So the final saliency map is given as:

$$S_f = [(1 - T) \times S_{BU}] + [T \times S_{TD}] \tag{2.8}$$

where S_f is the final saliency map generated by combining the top-down saliency map (S_{TD}) with the bottom-up saliency map (S_{BU}). This approach varies the contribution of each map on the overall saliency map manually by setting a value of T. Hence the approach is static, predefined and does not account for the dynamics of the situation in hand. However, the authors demonstrated the effectiveness of such a combination strategy in detecting balls in real world situations [21].

In [22], the authors did not use any time function, and hence simply combined the two maps linearly. In [109], the bottom-up saliency was totally ignored and only the learned weights were used to generate the top-down saliency maps. Another work by Rasolzadeh et al., [111], also used a linear combination of the two maps. However, the weights associated with the two saliency maps were dependent on the relative importance of these maps. They formulated a time varying tempo-differential equation to adjust the weight according to the following criteria:

$$\frac{dk}{dt} = -c \cdot k(t) + a \cdot \left(\frac{E_{BU}(t)}{E_{TD}(t)}\right), \quad k = \begin{cases} 1, & k > 0 \\ k, & 0 \leq k \leq 1 \\ 0, & k < 0 \end{cases} \tag{2.9}$$

where the E represents an E-measure map which is a formatted saliency map after computing the CSI indicator [109] described earlier in section 2.3.2.2. The weight k is weight used in the linear combination of the bottom-up and top-down maps. The parameters c and a are the concentration (devotion on the visual search task) and the alertness (susceptibility for any bottom-up attention) respectively. Equation (2.9) suggests that the top-down saliency factor come into consideration when E_{TD} is sufficiently greater than E_{BU}. When this is the case, the condition is equivalent to a situation when the target object is visible and the top-down saliency dominates the attention.

2.3.2.4 High-level features as top-down saliency

In this section, other top-down techniques based on high-level features, specialized descriptors, scene context and other variant approaches are discussed. High-level features have been rarely used in the past to define the top-down influence. In a very recent work [112], authors used high-level semantic information extracted from the input images to describe the top-down saliency. They argued that previous top-down saliencies used high-level descriptors tuned for particular objects. In case of natural images, it is difficult to have a descriptor for each object. Instead, a generalized high-level semantic information is used based on natural words similar to the visual word concept in text semantics.

In the training phase, the procedure for top-down starts with multi-level segmentation, followed by constructing a neighborhood similarity matrix based on some existing link-based similarity technique. Once these, high-level links are obtained, in the testing phase, a tagging mechanism is applied to each over segmented region, and finally the tag information of the segmented regions are added at multi levels to obtain a final likelihood measure of a region belonging to a foreground or background.

The bottom-up saliency is calculated through finding spatial, colour and complexity contrast using entropy measure of the segmented regions. However, the proposed model does not describe how the two saliency maps are combined. Furthermore, the purpose of the proposed model is to find salient regions and it is not meant for the task-based recognition problems. Finally, as described in section 2.3.2.1, many authors asserted that subjects set up the target in a more detailed and precise way and not in a semantic way [99].

In a more precise high-level scenario, the authors in [28] used a specialized face descriptor as top-down influence combined with other low-level features as bottom-up influence. They used the famous Viola and Jones algorithm to locate the faces. By testing their proposed method with a bottom-up fixation graph-based technique [91], they show fixation performance improvement in those images containing faces.

In images without faces, there was no significant degradation in false fixation points. Hence, they show that an inclusion of a high-level channel improves the fixation/saliency performance. Their work was an attempt to define the top-down influence in terms of precise high-level descriptors, in this case, faces descriptor, but in the context of fixation/saliency and not object recognition.

A popular model in top-down saliency is the SUN model based on natural image statistics [113]. This model describes the bottom-up saliency using self-information of a low level feature, and the top-down saliency as a likelihood function of the target feature selected amongst the low level features. The model combines both saliencies in a Bayesian framework and it is given as:

$$\log s_z = \underbrace{-\log p\left(F = f_z\right)}_{\text{(BU)}} + \underbrace{\log p\left(F = f_z | C = 1\right)}_{\text{(TD)}} \tag{2.10}$$

where z indicates the patch for which saliency is calculated, s is the saliency, F is the random variable representing a feature, and C is the random variable indicating the presence or absence of the target feature. The authors consider only the performance evaluation of their model in the absence of the top-down saliency term in Eq. (2.10). This models indicates a mutual information between the general saliency detection and the target detection.

Furthermore, when only bottom-up saliency is involved i.e., the first term in Eq. (2.10), the model implies the rarer the feature, the more salient it becomes. Hence the objective of this model is to maximize saliency by maximizing the self-information in case of free-viewing and maximize the self-information along with maximizing the log-likelihood function for a specific target feature.

The authors used two methods to extract features, one through the DoG filters and the other using ICA filter. Both these filters are learned from natural images. The model has a good performance when tested for fixation and saliency. However, no results were shown for target recognition.

In [114], the authors proposed an interactive attention system that combines the top-down with bottom-up in an interactive way to suppress unimportant salient regions generated by the bottom-up process. The bottom-up saliency is generated by extracting edges and symmetry features along with colour and intensity features from an input image. This is followed by applying ICA filters on the extracted features to get the final bottom-up saliency map. After that, an ART network is trained to interactively suppress the false salient regions through a human-feedback to the ART model.

Hence in the training phase, the ART model is trained to select the most salient regions in an image and suppressing the unimportant ones. In the testing phase, after finding the bottom-up saliency map, the input from the ART network act as a top-down influence by proving inhibition and excitation regions to the bottom-up saliency similar to those considered in [24, 106].

2.3.2.5 Top-down saliency and classification

In this category of models, the objective is to incorporate the knowledge of the object recognizer through classification to identify each salient region contain the target object or not. These models are dependent on the classifiers that are particularly tuned for detecting certain target objects in the image.

Previous work shows a strong interaction between the saliency and classification tasks. This interaction can be grouped into two main categories; saliency as ROI extractor for improving classification and learning the saliency through feedback from a classifier.

In the first category, the saliency mainly acts as a sampling procedure to extract ROI effectively and pass them to a classifier. The complexity of the classifier depends on the quality of the extracted ROI's. In many previous works, this domain was explored. In [71], the authors extracted the salient regions through the *Itti* model. This was followed by keypoint extraction from these salient regions suing SIFT and C2 features. These descriptors are matched with stored features from training images through a similarity measure based on Hungarian method. Furthermore, *k*-nearest-neighbors was used to find *k* best matches. Finally classification label was assigned. Results show a great improvement by including the saliency as ROI over some state-of-the-art classification techniques.

A similar approach was followed by Rutishauser et al. [31]. They investigated to what extent a pure bottom-up saliency technique can be useful in extracting meaningful ROI for the unsupervised learning of objects from unlabeled images. For movie shot classification, authors used saliency along with Support Vector Machine (SVM) to classify the shots from a video. Salient regions are extracted using contextual and geometric features. Results have shown a good performance in movie shot classification for all movie genres.

In another work, the ROI were extracted from two sets of feature pooling mechanism, a general bottom-up saliency-based followed by a high-level indoor scene based [115]. *Itti* model was used to discover visual-structures regardless of position importance (i.e., context). This was followed by another bottom-up saliency extraction of the salient regions using AIM. Furthermore, key point extraction and representation using SIFT descriptors was done on the bottom-up salient regions. For a better efficiency, the SIFT descriptors were applied to ICA for dimensionality reduction. Finally, SVM classifier has been used to classify the indoor scenes.

In the second category of classification based saliency, the bottom-up saliency maps were typically learned through a feedback from the classifier which was trained on human eye fixation datasets. However, the availability of reliable groundtruth fixation data and the computational resources of the existing techniques impose practical limitations.

A comprehensive review of different techniques in which saliency is learned is given in [116]. For instance, in [117], complex features tuned to specific class of objects were learned by a classifier and the initial bottom-up saliency belief was modified through these learned features. Furthermore, a discernment feature selection procedure was used before updating the saliency map through information content of the features.

2.3.3 Chapter summary

This chapter has covered various topics related to visual attention and its application in computer vision, particularly for target object detection. Researchers in the past have proposed techniques to accelerate the understanding of human visual attention system, mimicking some of its functionality and applying them to solve real time problems.

A detailed description of the human visual attention system from a biological perspective was given to have in-depth understanding of different areas of the brain responsible for performing a dynamic attentional shift from one location to another. Furthermore, the difference between two processes, namely a fast data-driven bottom-up process/saliency and a slower task driven top-down process/saliency was explored. Each process has its own merits and drawbacks when performing a high-level task such as object detection and recognition.

The chapter has also explored various performance measures commonly used in saliency and visual attention literature. These measures are based on comparing the generated saliency maps (bottom-up or top-down) with the groundtruth map (for salient object or fixation). Along with these measures, the chapter has provided some details about existing state-of-the-art bottom-up computational models for salient object detection.

Because the main objective of this thesis is to model top-down saliency and how to combine it with bottom-up saliency, the chapter has explored current techniques and approaches in modelling top-down saliency for target object detection. Very few computational techniques exist that describe the fine details of this process compared to a well studied and explored bottom-up process.

Four different types of contribution are made in modeling top-down process. First is the modulation of bottom-up features through learned weights that are specifically tuned for a particular task. Second, is the use of various high-level features such as face detectors, text detectors and target specific descriptors to model top-down saliency. Third, is to apply classification and object recognition techniques specifically for those visual attention applications that involve visual search for the target object. Finally, how to combine bottom-up with top-down process to maximize the detection of the target with minimum resources.

The limitations of the existing approaches for top-down saliency that constitutes the motivations are summarized below:

1. Top-down models that perform target object detection through appropriate weighting of low-level features fail to adapt to scenes with content variation. This is because such models do not utilize any high-level information of the scene while learning these weights. As a result, a contextually based feature weighting that extracts high-level gist information from a scene is required to improve the accuracy of detecting the target object in complex scenarios.

2. Top-down models that implicitly use target specific features tend to be computationally demanding as specialized detectors are used for this purpose (e.g., pedestrian or text detectors). Furthermore, these models while considering target features do not associate them with contextual information. As a result, a model is needed that uti-

lizes low-level features instead of specialized detectors for better efficiency. In addition, contextual information combined with target features have not been considered previously while modelling top-down saliency.

3. Most of the models that combine top-down and bottom-up saliency treat both influences separately and do not consider which type of saliency is important for a task. This results in performance degradation as combining both saliencies does not always result in the best detection performance. Hence, a model is needed that can interactively and dynamically decide which saliency or feature to combine that would maximize the detection accuracy.

4. Existing models that combine both saliencies for target object detection learn a single combination rule which is applied to all the images in a particular dataset. This static approach does not always provide the optimum combination solution for each image in a dataset. A dynamic framework is needed that provides a run-time evaluation mechanism to decide upon which feature maps to exclude from the combination process on image basis.

Chapter 3

Contextual Top-Down Feature Weighting

3.1 Chapter introduction and motivations

A detailed description of the first biological inspired visual attention model proposed by Itti et al. [1] was presented in the previous chapter. Various extensions to this model have been proposed, mainly to weight the features before integration to yield the final saliency map [20–24].

The main challenge in feature weighting is how the weights can be dynamically assigned to the features for a specific task. While different weight learning approaches have previously been adopted, most of these approaches suffer from being static. This means that the learned weights could only work in similar types of examples/images and would fail when the examples/images are different in content (e.g., objects, background, etc.).

It has been shown experimentally that the inclusion of contextual information improves the efficiency and accuracy of discriminating the target object from the background distractors [30, 118, 119], though the context idea has not previously been applied to *TD* saliency and feature weighting.

In this chapter a mechanism that utilizes the contextual information of an image to dynamically assign appropriate weights to the *BU* features is introduced. Such weighting allows the demands of a particular task to modulate the bottom-up responses which results in a top-down model particularly tuned for that task. The *BU* features used are the typical low-level features commonly used in attention models such as orientation, intensity and colour. The inclusion of contextual information for *BU* feature weighting has not been considered in previous works. Hence, such dynamic feature weighting for target object detection constitutes the main contribution of this chapter.

The proposed model referred to as top-down contextual weighting (*TDCoW*) mainly consists of three functional parts. Each part has its own significance in building the proposed model, however only the part related to contextual information extraction, clustering and matching constitute the major contribution of this chapter. Justifications are provided in the corresponding sections for the inclusion of these parts in the overall model.

The first part is concerned with the bottom-up approach used for saliency map generation. This itself contributes towards having a better bottom-up saliency generation mechanism than existing state-of-the-art bottom-up techniques. However, as mentioned above, this does not constitute the major contribution of the chapter, but rather contributes towards building a better top-down saliency model.

The second part of the model is concerned with computing the weights that will be assigned to feature and conspicuity maps. An approach based on Jensen-Shanon Divergence *JSD* is used to allow better weight representation than previously proposed signal-to-noise *SNR* approach. Again this part only provides a better way to compute the weights but does not play a direct role in the contextual based dynamic weight assignment.

The final part represents the core contribution of this chapter. It consists of the actual construction of a contextual descriptor of images, contextual clustering based on the similarity between these descriptors and matching a contextual descriptor of a novel image with the clusters' descriptors for weight assignment.

3.1.1 Bottom-up model modification

Our proposed model uses a similar structure to the *Itti* model for bottom-up saliency map generation. However, a number of modifications are necessary before proceeding with the contextual information incorporation into our model. These modifications are essential to generate better saliency maps than those generated by the *Itti* model for the purpose of target object detection. In addition, the justification for each modification step is provided below:

1. **Modification 1:**
   To build the final *TD* model, the first step is to have a set of good features to perform the weighting on. Hence, instead of using the traditional *Itti* features, a richer set of low to medium level features that can be useful in the target object detection task are used to expand the domain of feature weighting. These features will have a direct impact on the quality of the generated saliency map. Because the effectiveness of the TD model depends on the feature weights, which in turn depend on the feature themselves, a good set of feature is likely to establish a good TD saliency model.

2. **Modification 2:**
   The features are computed over a single scale instead of using multiple scales and different Gaussian filters to speed up the process of generating the final saliency map. In addition, the multiple scale approach of the *Itti* model is necessary because of its use of contrast features. This is not appropriate for many of the features used in the proposed approach.

3. **Modification 3:**
   The contrast based centre-surround mechanism is replaced by an information theoretic approach called *Information divergence* proposed in [120]. The centre-surround proposed in *Itti* model is a biological mechanism that is implemented through multi-scale difference of Gaussian operation. Since we do not compute features on multi-scale, a different computational based mechanism is required. The information divergence based centre-surround mechanism is not only more computationally efficient than the contrast based approach but also has achieved a very good performance in saliency detection comparable to state-of-the-art techniques for single scale features [120].

4. **Modification 4:**
   The final conspicuity map integration is done through multiplication operation rather than the summation as it has been proven to yield a higher precision in detecting the target object or region of interest [121]. This implies that salient targets must exhibit a range of features, rather than just having values for one feature.

3.1.2 Chapter objectives and overview

The overall objective of this chapter is to show that a better weighting of features can be achieved when contextual information is considered during feature weight learning. To achieve this objective, the following tasks are performed,

1. **Calculation of the feature weights using the *Jensen-Shanon Divergence* (*JSD*)**
   Weight calculation is an essential part of the feature weighting model. Previously the *SNR* approach was used for weight calculation [22]. However, *SNR* calculation gives unbounded weight values. In addition, *SNR* is calculated by finding the ratio between the

mean saliency values of the target region and that of the background region. This could lead to inverted weight assignment to a feature map in situations when the target region has lower intensity values than the background. A better approach is to use the difference between the distributions of both regions. *JSD* is chosen as the appropriate measure as it is bounded.

2. **To construct a contextual descriptor for an image**
   Different contextual descriptors were previously used to provide a holistic description of images such as the envelope gist descriptor [2] and the bag-of-visual-words based descriptor [122]. These descriptors tend to be computationally demanding particularly when performing descriptor matching. A new contextual descriptor is proposed that is constructed through probability density estimation of some of the selected low level features from the modified version of the *Itti* model.

3. **Clustering of training images based on their contextual information**
   A contextual descriptor is generated for each image and then similar contextual images are clustered during the training phase. Distinct weights are computed for each cluster and used in the testing phase. For a new test image, the context information is extracted and matched to the context of each cluster. The feature weights of the best-matched cluster are assigned to the test image.

The following steps briefly explain how to assign weights to the feature and conspicuity maps dynamically of a novel image using the proposed model *TDCoW*:

1. Split the dataset into training and testing sets.

2. For each training image, compute its feature and conspicuity maps

using the proposed bottom-up approach (refer to section 3.3 on how to generate these maps using a modified version of the Itti model).

3. Using the groundtruth maps of the training images, highlight the target and not-target regions on the feature and conspicuity maps obtained from the previous step.

4. Compute the weights of the feature and the conspicuity maps with the help of the highlighted target object region using the *JSD* approach (refer to section 3.4.2 for details). Note that at this stage, for each training image, we have two sets of weights, the feature map weights (a single weight value for one feature map) and conspicuity map weights (a single weight value for one conspicuity map).

5. Extract a contextual gist descriptor for each training image (refer to section 3.4.1 for details).

6. Construct clusters using k-mean clustering technique. This is performed based on a distribution similarity measure between these descriptors (refer to section 3.4.3) for details. Each cluster will have a centroid descriptor representing the gist content of the images belonging to that cluster.

7. The feature and conspicuity map weights of individual images belonging to the same cluster are combined by averaging their values. In this way a single set of feature and conspicuity maps weights are generated for that cluster. This step is repeated for all the clusters.

8. For In order to dynamically assign feature and conspicuity map weights for a novel image from the test set, initially its contextual gist descriptor is constructed.

9. The contextual descriptor of the novel image is compared with that of the stored centroid clusters that are obtained in the training phase.

The comparison is performed using *JSD* approach. The feature and conspicuity map weights of the best matched cluster are chosen.

10. The selected weights are used on the feature and conspicuity maps to construct the final saliency map through the saliency generation process described in section 3.3. In this way, weights are dynamically assigned to the corresponding maps of the image (i.e., top-down feature weighting) that would maximize the detection of the target object.

3.2 The proposed model: top-down contextual weighting (*TDCoW*)

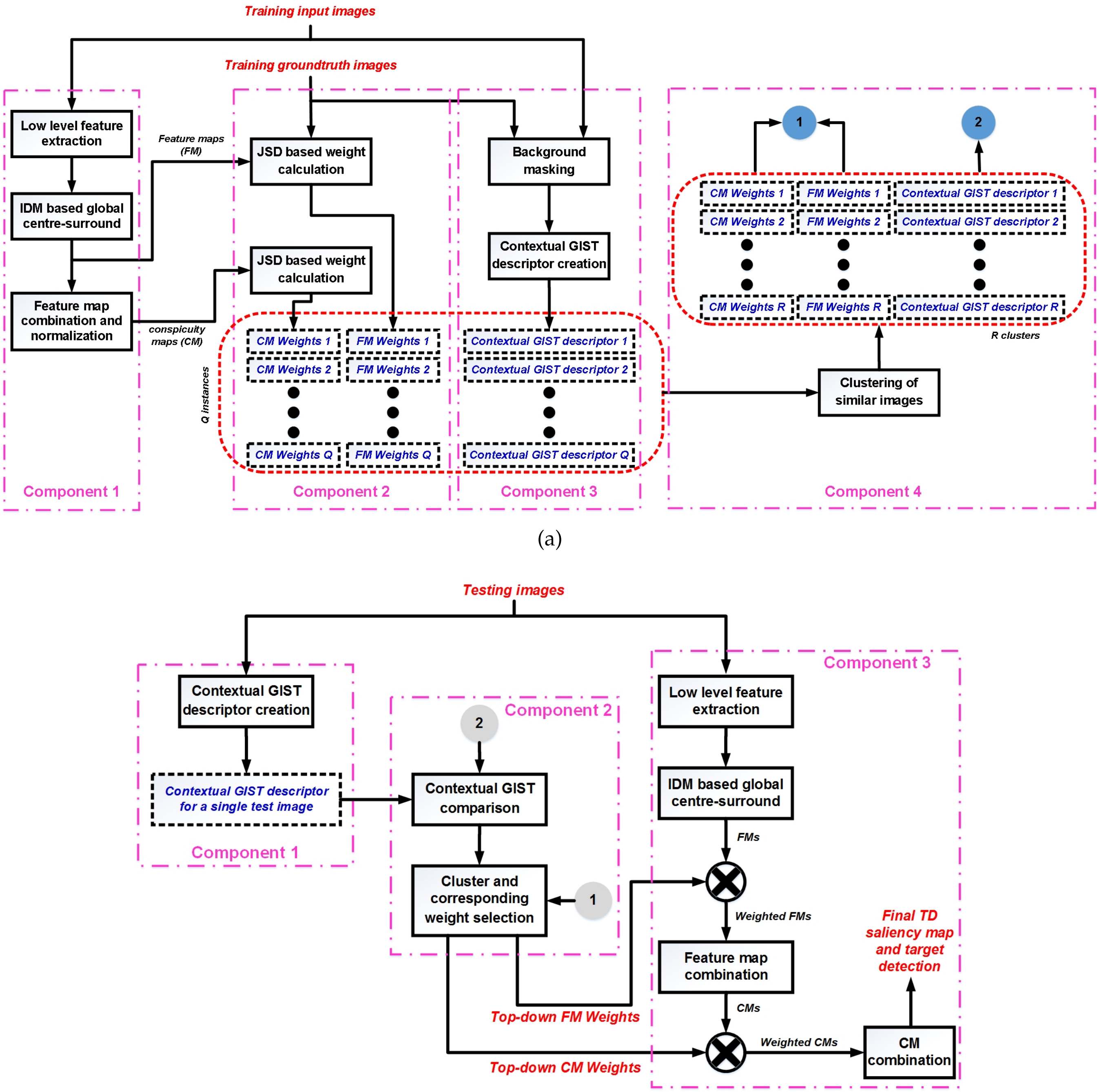
In the proposed model the contextual information represents only the gist of the image. Hence, gist and context are used interchangeably. The proposed model, referred to as Top-down Contextual Weighting (*TDCoW*), is used in two phases, the training and testing phases. As shown in Fig. 3.1(a), the first step of the training phase (labelled 'component 1' in the figure) starts with feature extraction followed by a centre-surround measure called *Information-Divergence Measure* (*IDM*) to yield feature maps (CM). Note that we use the same terminology that was used in the original *Itti* model to describe various components of the saliency generation model. In the *Itti* model, the feature maps represented the centre-surround differences at a number of scales. In this proposed model, feature maps refer to simple features that are extracted from an image at a single scale followed by a different type of centre-surround measure applied on a single scale feature. More about this centre-surround is discussed in section 3.3.2. Again just to be consistent with the notation followed in the *Itti* model, when similar type of features are combined together (e.g., colour features) followed by a normalization process, conspicuity maps (CM)are generated. The FMs and CMs are generated for Q instances in the training set.

In the second step labelled as 'component 2', the computation of the feature weights for each training image takes place. Two sets of weight are computed for each image, the weights computed from the FMs and the weights computed from the CMs. The weights are calculated by finding the *Jensen-Shanon Divergence* (*JSD*) of the target object with respect to the background. The details about how this calculation is performed is given in section 3.4.2.

The third step labelled as 'component 3' in the training phase involves contextual information extraction. The contextual data of the entire image is extracted except from the target region. This is because target objects are not considered to be part of the overall gist of a scene (mostly the background represents this information). To exclude the target object from the contextual computation, the image is masked with the groundtruth map so that the context is only extracted from the background region. This is followed by the creation of the feature descriptor from the background region. The descriptor acts as a context identifier for the image which can later be used for contextual matching. Hence each training image is associated with FM weights, CM weights and a contextual descriptor as shown in Fig. 3.1(a).

In the fourth step labelled as 'component 4' of the training phase, grouping of the training images into R clusters is performed. The grouping is done according to the contextual similarity amongst images using unsupervised k-mean clustering. For each cluster, a contextual descriptor is calculated as the average of the individual contextual descriptors of the images belonging to that cluster. Similarly, the weights of the individual images in a cluster are averaged to yield consolidated weights for that cluster. Hence, in the figure (see 'component 4' of the training phase), the contextual GIST descriptor represents the GIST of that cluster while CM or FM weights are the weights associated with that cluster.

The second phase of the model is the testing phase in which a new


Training input images
Training groundtruth images
Low level feature extraction
IDM based global centre-surround
Feature map combination and normalization
Feature maps (FM)
conspicuity maps (CM)
Q instances
JSD based weight calculation
JSD based weight calculation
Background masking
Contextual GIST descriptor creation
CM Weights 1
FM Weights 1
CM Weights 2
FM Weights 2
CM Weights Q
FM Weights Q
Contextual GIST descriptor 1
Contextual GIST descriptor 2
Contextual GIST descriptor Q
Clustering of similar images
1
2
CM Weights 1
FM Weights 1
Contextual GIST descriptor 1
CM Weights 2
FM Weights 2
Contextual GIST descriptor 2
CM Weights R
FM Weights R
Contextual GIST descriptor R
R clusters
Component 1
Component 2
Component 3
Component 4
Testing images
Contextual GIST descriptor creation
Contextual GIST descriptor for a single test image
Component 1
Component 2
2
Contextual GIST comparison
Cluster and corresponding weight selection
1
Component 3
Low level feature extraction
IDM based global centre-surround
FMs
Weighted FMs
Feature map combination
CMs
Weighted CMs
CM combination
Final TD saliency map and target detection
Top-down FM Weights
Top-down CM Weights

(a)

(b)

Figure 3.1: The proposed *TDCoW* model: (a) Training phase (b) Testing phase.

image is processed. As shown in Fig. 3.1(b), the process begins by creating a contextual descriptor for the test image ('component 1')). However, now the context must be calculated over the whole image without any target masking due to unavailability of the groundtruth. This will cause some perturbation in the contextual descriptor but it is anticipated to be minor as long as the ratio of target region to the background is small. This contextual descriptor is compared with the centroid descriptor of each learned cluster using an appropriate information theoretic distance metric as shown in 'component 2' of Fig. 3.1(b). The cluster with the lowest distance corresponds to the best match for the test image. Accordingly, the matched cluster's weights are selected to be used as the *TD* weights for the test image. In this way, appropriate weights are assigned to the FMs and CMs of the test image.

If the contextual information of a set of images is diverse, the clustering will still take place but now the clusters' contextual information will have high variance with respect the contextual information constituting that cluster. In order for this clustering to be meaningful, a good contextual agreement between the images of a dataset is essential.

The final component represents the process of generating a saliency map as before, but now the FMs and the CMs are weighted by the weights that were learned in the training phase. A more detailed description of each step in the training and testing phases is described in the next section.

Note that multiple sets of weights and contexts are learned during the training phase, one for each cluster. The test phase acts as a context template matching process that results in a best possible weight assignment to the FMs and CMs of the test image. Because the weights are assigned to the test image based on its contextual information, the weights become dynamic whenever the contextual information changes. Hence, for two images, if their context is different, they will be assigned different weights. Unlike previous approaches in top-down weight learning where a unified set of weights are learned for all the images in the dataset [22, 106], the weight is dynamically assigned here depending on the context of the image.

3.3 Saliency map generation process

3.3.1 Bottom-up feature extraction

The initial step in generating the saliency maps is feature extraction. Eight features are extracted as shown in Fig. 3.2. These features are colour (C), intensity (I), orientation (O), contrast (Co), centre-bias (Cb), principal component analysis features (PCA), edges (Ed) and frequency based features (MSS) with each feature category having sub-features. The features range from low-level features such as colour and intensity to higher level features such as edges and *PCA* features.

There are two reasons to have many extracted features. First, some high-level features such as *PCA*, edges and frequency are assumed to give insight to valuable information about the structure and behavior of the image. Such information can be very useful for better saliency estimation and target detection. Secondly, some targets may require additional features or different subsets of features in order to detect them. Itti's basic features might not be sufficient for target detection. On the other hand, there is no specific number or type of features that are assumed to be sufficient for a general target detection. As a reasonable set of features for target detection, the above mentioned features are used which is a combination of low, high and efficiently computed features.

3.3.1.1 Colour features

The colour feature has four sub-features; red (r), green (g), blue (b) and the quantized colour feature (q). The quantization is performed as follows [92]: assuming an input colour RGB image is of size $H \times W \times 3$ with eight bits colour depth, the first step is to quantize the colour range by

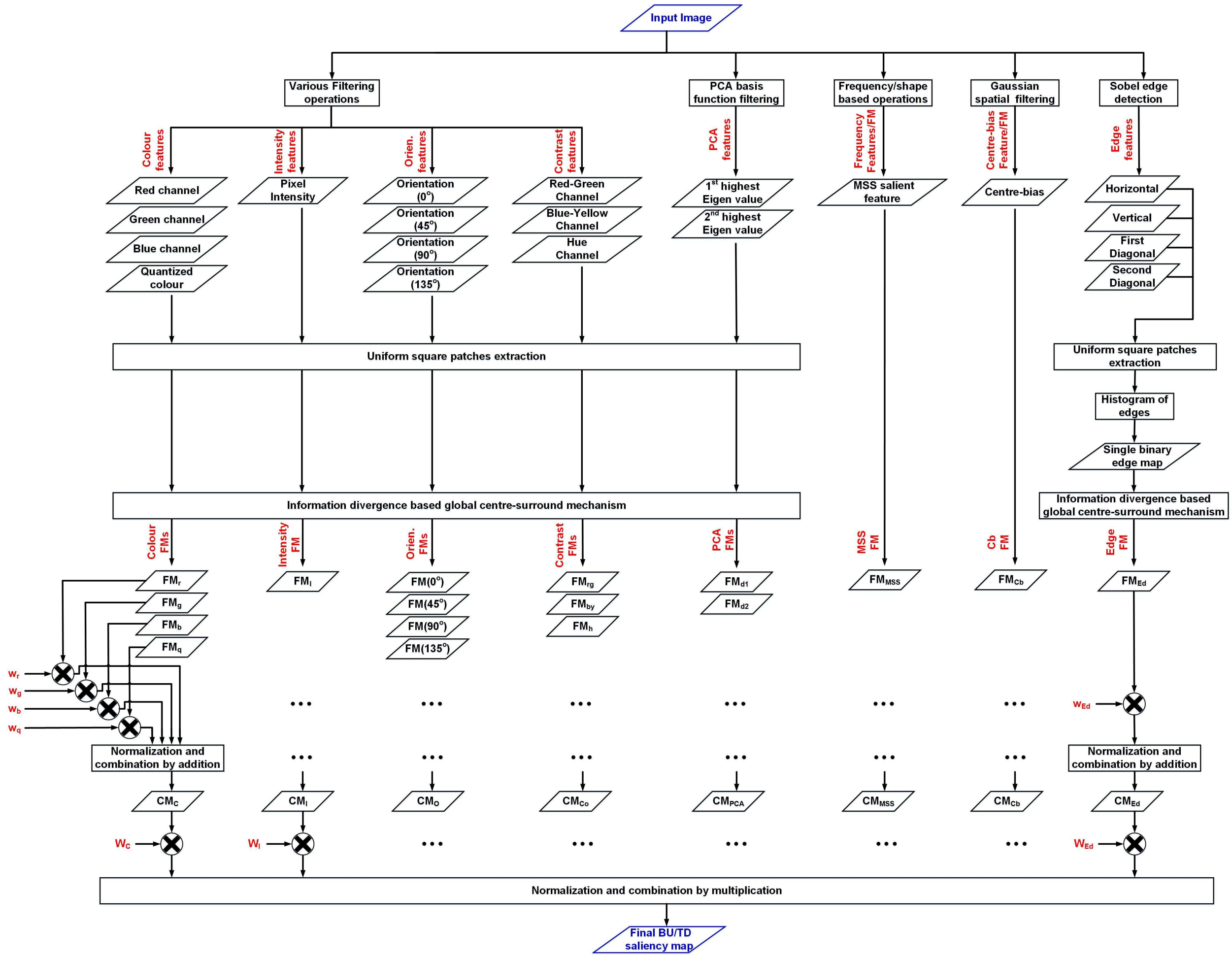


Figure 3.2: Feature extraction and saliency map generation procedure.

selecting 12 uniformly distributed levels for each colour channel. This will yield $1,728$ different possible colours. Furthermore, only 5% of the most occurring colours in natural images are retained. This is done by observing the most frequent colours from a large database of natural images. The images are quantized with these levels yielding a palette of 85 colours. The quantization is done to reduce the histogram space from 256^3 to only 85 for the subsequent *IDM* calculation.

The intensity feature has a single sub-feature denoted as I and the orientation features are extracted at $0^o, 45^o, 90^o$ using Gabor filters. Contrast sub-features are also computed as they have good performance in saliency detection [76,92,94], which includes the Red-Green (rg), Blue-Yellow (by) and Hue (h). These sub-features are given as:

$$\begin{aligned} rg &= \frac{r-g}{\max(r,g,b)} \\ by &= \frac{b-\min(r,g)}{\max(r,g,b)} \\ h &= \frac{180}{\pi}\arctan\left(\frac{\sqrt{3}(g-b)}{2r-g-b}\right) \end{aligned} \tag{3.1}$$

It is worth mentioning here that the colour features are object-level features and more suited for object detection whereas colour contrast is more effective for saliency detection. This is the reason for including both types of features.

3.3.1.2 PCA based features

The next feature is the *PCA* based features. *PCA* is a statistical approach that transforms a set of correlated observations/features into orthogonal uncorrelated segments called principal components (PCs). Irrelevant details and noise are neglected while finding these PCs. The obtained PCs are assumed to describe important features contained in the images. Previously, *PCA* was used for extracting useful features for salient object detection [94,113,120]. One way to implement *PCA* on the input image is to consider square patches of the input image and then to extract the PCs of the patches separately [120]. However, in this work *PCA* is applied colour wise rather than patch wise for better efficiency as the number of patches $L \gg 3$, where three colour channels are considered.

An input colour image of size $H \times W \times 3$ is reshaped so that each colour layer is transformed into a column vector of length $H \times W$, denoted $\mathbf{x}_i$. After concatenating the three layers column wise, we get a reshaped layer matrix $\mathbf{X}$ representation of the image such that $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3]$.

The mean value of each column in **X** is subtracted from the corresponding column. This is followed by the computation of the covariance matrix of **X** and then the eigenvectors. Each eigenvector represents one of the PCs of a total of three PCs denoted as d_1, d_2 and d_3. These components are ordered according to the magnitude of their eigenvalues such that d_1 is associated with the highest eigenvalue. In the experiments, it has been observed empirically that only the first two components prove useful.

3.3.1.3 Edge feature

The next feature is the edge map which is extracted in vertical, horizontal and diagonal directions. Although there are four sub-features for the edge feature, these are considered as a single sub-feature denoted as *'Ed'* for the reason mentioned later in this section.

3.3.1.4 Centre-bias feature

A centre-bias factor is added as an additional feature which is represented by a 2-D Gaussian function centered at $(\frac{W}{2}, \frac{H}{2})$ and controlled by the standard deviation value of the function. The centre-bias behavior models human prior knowledge that the centre of photographs tends to contain salient targets. However, in the test examples, this factor exhibits low weight value, which suggests that the datasets used did not exhibit strong centre-bias.

3.3.1.5 Frequency based feature

The final feature called maximum symmetric surround denoted as (*MSS*) is basically a saliency detection technique proposed by Achanta and Sabine [86]. The idea behind their approach was to use different band-pass filters on a local region of the image by varying the low-frequency cut-off. The variation is based on the location of a pixel with respect to a symmetric local region surrounding that pixel. By making assumptions about the scale of the object to be detected based on its position in the image, those pixels that are positioned far from the salient object's boundaries need small cut-off frequencies to be successfully detected. Through this approach, low-frequency components and majority of the high frequencies of the object are retained.

Once the features are extracted, the FMs are generated by calculating the *IDM* of various patches of the image as proposed in [120] and elaborated below. *Information-Divergence Measure* is a centre-surround the mechanism that exploits the element of surprise by finding the divergence of information between various regions of the image. Hence, it is responsible for generating the FMs that highlight the possible salient regions within a feature.

3.3.2 Information divergence based centre-surround

This procedure is the same for all the sub-features except for the edge and the centre bias. The process starts by dividing a feature image into smaller regions by uniformly segmenting it into square non-overlapping patches of size $n \times n$. The total number of complete patches is $L = \lfloor H/n \rfloor \times \lfloor W/n \rfloor$. For instance, if an image has the dimension of 12×12, and if the patch size is 5×5, then we will have only $L = 4$ complete patches which are indexed as $i = 1, 2, 3, 4$. These patches are denoted by $p_i^j(k)$ where $i = 1, 2, \ldots, L$ represent the patch index, k is the feature notation for which the FM is to be generated and j is the sub-feature notation for the respective feature. For instance, $p_1^r(C)$ is the first patch for the sub-feature red belonging to the colour feature.

The *IDM* is calculated for each patch by finding the divergence of the distributions between two patches for a particular sub-feature. The first patch (centre patch) is one of the patches from $p_i^j(k)$ where as the second patch (surround patch) is the collection of the remainder of the patches as illustrated in Fig. 3.3. If G and S represents the kernel density estimated (KDE) distributions for the feature values of the centre and surround re-

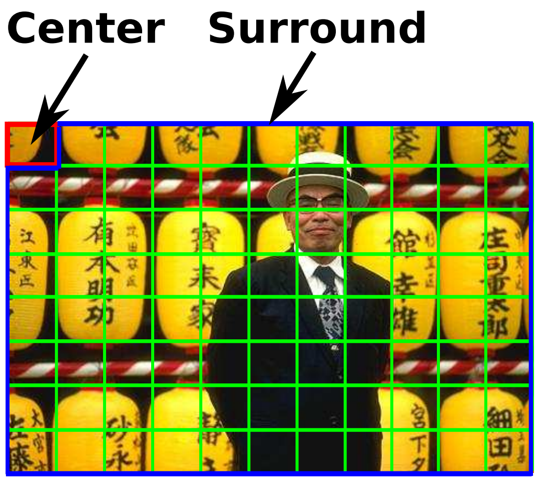


Figure 3.3: Global centre-surround patches. The patch within the red square represents an active centre patch selected amongst the green squares. The collective surround patch is highlighted by the blue region.

gions respectively, a patch saliency is found by:

$$IDM(i,j,k) = \sum G_i^j(k) \log\left(\frac{G_i^j(k)}{S_i^j(k)}\right) \tag{3.2}$$

The centre-surround concept introduced in the *Itti* model mimics the visual attention behavior in humans where visual neurons are highly sensitive in a small concentrated region at the attention location (the centre) and weaker in the surround region concentric to the centre. Itti modelled this behavior using multi-scale difference using the difference of Gaussian (DoG) filters were higher scales represent the centre (fine details) and the surround represents lower scales (coarse detail) [1]. In the proposed approach, the concept of centre-surround is equivalent to the one proposed in the *Itti* model, but with the following exceptions.

First, the mechanism is performed on a single scale. Secondly, it uses a more information theoretic approach using *IDM* instead of difference. Third, the surround region does not necessarily be concentric to the centre as shown in Fig. 3.3. Hence, it is assumed that once a patch is attended to (centre logically), the rest of the regions of the image (in a non-concentric manner) become the surround. Finally, we go one step ahead of the notion of centre-surround by using *IDM*. Unlike Itti's centre-surround approach that assumes higher sensitivity of the centre to the surround for any attended location, the *IDM* based approach, on the other hand, assumes that the centre is more sensitive than the surround only when it carries information that is highly diverged from the surround. If it is not, then the centre is not sensitive or important to attend. As we will see in the result section, this has a positive impact on the saliency map either detection of the salient object or the target object.

For the centre bias, the FM is the feature image itself and there is no need to evaluate the *IDM*. In addition, for the edges, the *IDM* is calculated in the same way but not directly on the sub-feature edge images. Initially a histogram of edge orientation is evaluated for each patch from the four sub-feature binary edge images. This is followed by *IDM* calculation but now on discrete edge histograms. The reason for computing *IDM* on the histogram of edge orientation rather than the binary edge map directly is because the latter does not carry any useful information about the edges importance in the image. Instead, the orientation of these edges carry more useful information about the distribution/frequency of such edges in the image.

3.3.3 Conspicuity map combination

The CMs are generated by summing the weighted sub-features within a feature. Finally the weighted CMs are combined by multiplying them together to yield the final saliency map. Multiplication of CM is better than adding them when combining different types of features as the maps tend to have different spatial distributions and thus common regions of interest from each map are needed. The map integration procedure can be summarized by the following equation:

$$
\begin{aligned}
\mathrm{SM} &= \mathcal{N}\left(\prod_{k=1}^{8} W_{\mathbf{u}(k)}\left(\sum_{j=1}^{\mathbf{v}(k)} w_j^{\mathbf{u}(k)}\left(\mathrm{FM}_j^{\mathbf{u}(k)}\right)\right)\right) \\
\mathbf{u} &= (C, I, O, Co, PCA, Cb, Ed, MSS) \\
\mathbf{v} &= (4, 1, 4, 3, 2, 1, 1, 1)
\end{aligned}
\tag{3.3}
$$

where $w_j^{\mathbf{u}(k)}$ is the weight associated with each FM and $W_{\mathbf{u}(k)}$ is the CM weight. The tuples $\mathbf{u}$ and $\mathbf{v}$ represent the feature symbol tuple and the number of sub-feature tuple corresponding to each feature respectively. Note that $\mathcal{N}$ indicates a normalization step before acquiring the final saliency map to promote very strong peaks and suppress the rest. This step is essential to obtain a saliency map with only the most dominant salient region being highlighted.

3.4 Contextual feature representation and feature weighting

This section describes how contextual information is extracted and then used to assign dynamic weights to the low-level features. In addition, an effective weight calculation procedure is explained that depends on the *JSD* between a target and rest of the image.

3.4.1 Contextual descriptors

Contextual based image descriptors are commonly used in classification problems [71, 123] where such a descriptor gives a global representation of a scene content. For instance, the descriptor proposed by Oliva and Torralba [2] provides a perceptual distribution of various parameters in a scene such as naturalness, roughness and ruggedness. However, one drawback of such a descriptor and many others is that they are computationally demanding. As a result, a more compact global descriptor is required to improve the efficiency of the overall system, particularly when performing descriptor matching.

As a result, a contextual descriptor of distributions for an image is created using colour, intensity, orientation, contrast and *PCA* features. The distribution is estimated using KDE for a fixed number of sample points. Hence, the size of the descriptor depends on the KDE number of estimation points. A large number of sample points corresponds to large descriptors that will impose relatively high computational demands for contextual matching. At the same time, we want to avoid under-sampling which can produce inaccurate results. Empirically the number of sample points is set to 1000. Figure 3.4 shows the complete procedure for generating the contextual descriptor.

3.4.2 Feature weighting

Previously in [22], weights were evaluated by calculating the signal (the target) to noise (distractors) ratio from the FMs. There are two shortcomings of the *SNR* weighting mechanism. First, the weighting information is extracted from the sub-features directly and not from the FMs. The value of the weight is highly dependent on how the FMs are generated. For instance for a particular sub-feature, if the corresponding FM is generated inaccurately (the target region is less highlighted than the background region), then according to the *SNR* Eq. (2.6) and Eq. (2.7), such an FM will receive a low weight whereas it should be assigned a high value. Hence, a poor FM generation algorithm could result in a wrong weight assignment to the feature when *SNR* approach is used.

Secondly, the *SNR* is calculated by finding the ratio between the mean pixel intensity values of the target region and that of the background region. Again this could lead to an inverted weight assignment to a CM in situations when the target region has lower intensity values than the

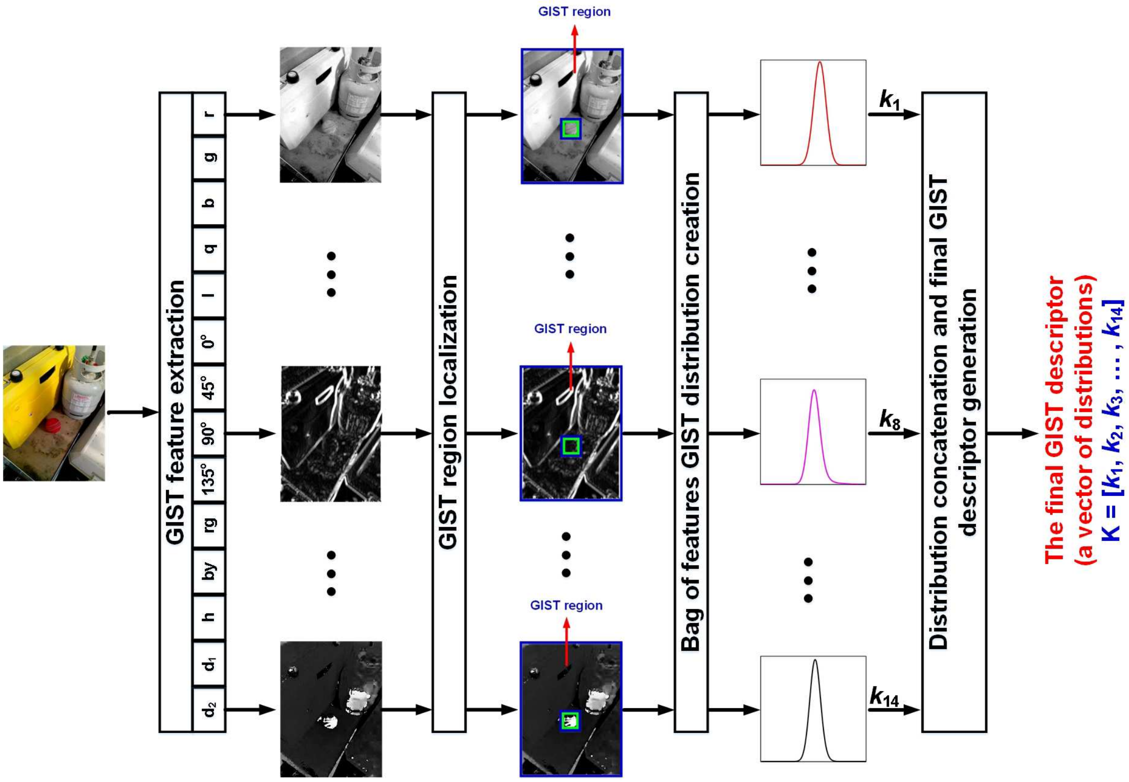

GIST feature extraction
r
g
b
q
I
0°
45°
90°
135°
rg
by
h
d1
d2
GIST region localization
GIST region
GIST region
GIST region
Bag of features GIST distribution creation
k1
k8
k14
Distribution concatenation and final GIST descriptor generation
The final GIST descriptor
(a vector of distributions)
K = [k1, k2, k3, ..., k14]

Figure 3.4: An example of contextual descriptor construction. From left to right: input image, extracted features, masking regions (the green region is excluded and the blue region is the gist region for which a descriptor is created), distributions of the gist regions, concatenation of the distributions and the final contextual descriptor vector.

background within the CM. This is because the ratio/difference is taken between two single values (i.e., the mean values of the two regions).

To overcome the problems associated with the *SNR* based approach, a distribution based weight calculation approach is adopted using *JSD*. *Jensen-Shanon Divergence* is a bounded dual version of the *IDM*. The divergence between the target region having a distribution h_T and the background region with distribution h_B is calculated directly from the subfeatures rather than the FM to avoid the involvement of the FM generation algorithm. The weight is calculated as follows:

$$\begin{aligned}
JSD\,(T||B) &= \frac{1}{2}\left(Z_{\text{IDM}}\,(T||M) + Z_{\text{IDM}}\,(B||M)\right)\\
h_M &= \frac{1}{2}(h_T + h_B)\\
Z_{\text{IDM}}\,(T||M) &= h_T \log_2\left(\frac{h_T}{h_M}\right)\\
Z_{\text{IDM}}\,(B||M) &= h_B \log_2\left(\frac{h_B}{h_M}\right)
\end{aligned} \tag{3.4}$$

where h_M is the histogram of the intermediate region M. The terms $Z_{\text{IDM}}\,(T||M)$ and $Z_{\text{IDM}}\,(B||M)$ are the target/intermediate and the background/intermediate regions' *IDMs* respectively. Because the *Jensen-Shanon Divergence* computation is a symmetrized and smoothed version of the KL-divergence between the target and background regions, it requires an intermediate distribution denoted by h_M and to go with the region convention, the intermediate term M is given a region notation that is arbitrary.

It is worth mentioning here that a statistically high *JSD* value for a particular sub-feature suggests that the target region is highly different from other regions of the image. This indicates the importance of that sub-feature for the target detection and should be assigned a high weight. As an example, Fig. 3.5 shows how the weights vary due to the statistical information difference of the target and the background for some sub-features and this can be reflected in the corresponding weighted FMs. For instance, the distributions of the target, background and the intermediate region M for the red/green sub-feature have high variation. Hence, the assigned weight to this sub-feature is relatively high (i.e., 0.64). This can be observed by the corresponding FM which highlights the target red ball better than the other two sub-features.

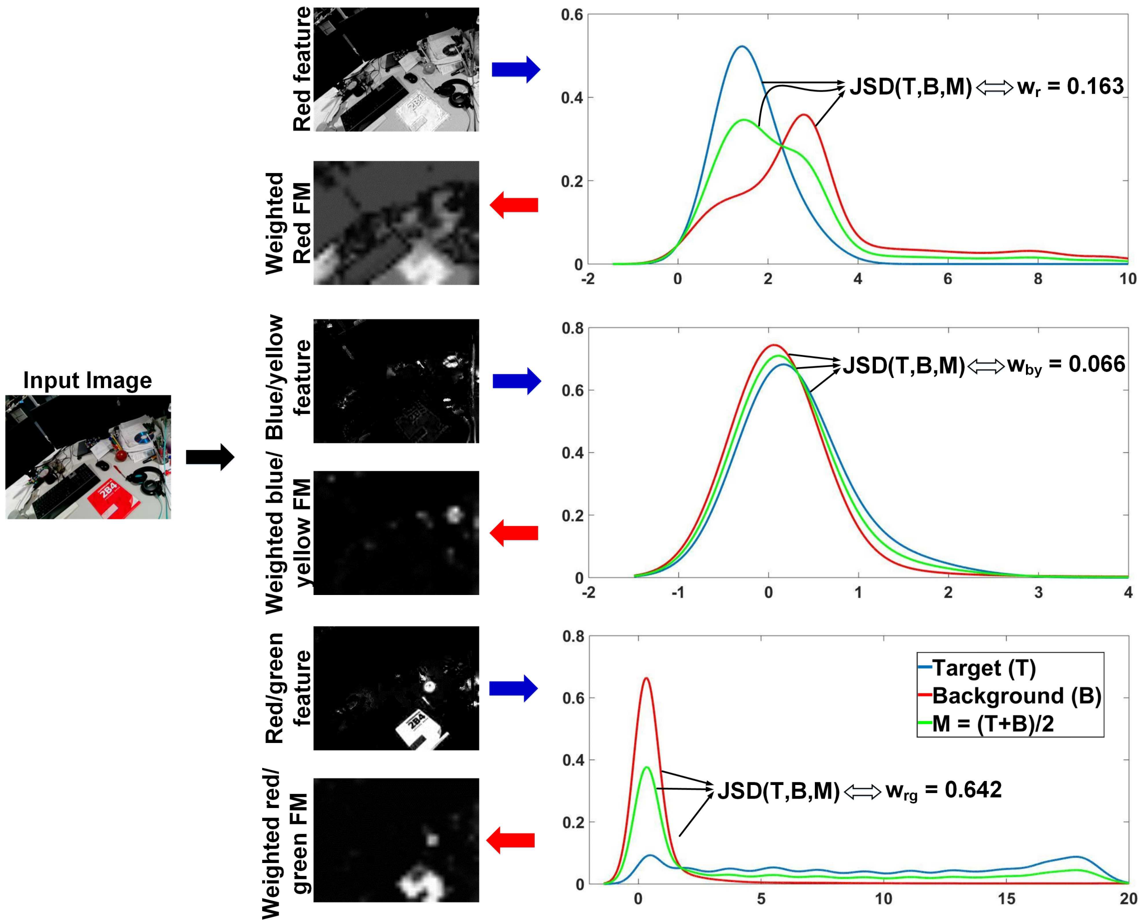


Figure 3.5: *JSD* based top-down weight calculation example for three sub-features, red, blue/yellow and red/green that can be seen to the left of the blue arrows. Distributions of target (T), background (B) and the intermediate region (M) for the respective sub-features are shown on the right column. According to the variation in distribution, a weight value is calculated using Eq. (3.4) as 0.163, 0.066 and 0.642 respectively for the three sub-features. The images on the left of the red arrow shows the impact of the weights on the FMs. For this example, the best map in detecting the target red ball is the red/green feature which is assigned the highest weight value compared to the other sub-features.

3.4.3 Clustering and contextual matching

The main objective of *TDCoW* is to allocate the test image appropriate weights according to the similarity of its contextual contents with that of the training images. This could be done by matching the contextual descriptor of the test image with the training images. However, increasing the number of training images would then increase the processing time for matching. As a result, an attractive alternative would be to cluster similar images together to reduce the search space for matching according to the contextual similarity.

The clustering approach is similar to the classical *k*-means clustering except for the distance measure calculation. The clustering is done on the contextual descriptor of the training images over several iterations. After every iteration, clusters are created by measuring the *JSD* between descriptors. The reason for using *JSD* and not the traditional Euclidean distance is that the distance is measured between distributions. Furthermore, once clusters are created after a single iteration, a mean descriptor for that cluster (called a centroid descriptor) is re-calculated from the individual contextual descriptors of the images belonging to their respective clusters. This is achieved by considering the joint distribution of the individual descriptors of the cluster.

After the final iteration, the centroid of each cluster represents the final contextual descriptor for the corresponding cluster. Furthermore, the learned weights of the individual images of a cluster are averaged to yield a single set of weights for that cluster. Hence, at the end of the clustering process, *R* number of descriptors and weights are produced, one for each cluster.

When a test image is given, the ultimate objective is to assign appropriate weights to the features to produce the final *TD* saliency map. The following steps are considered:

1. Creating a contextual descriptor for the test image by following the

same procedure applied to a training image, as explained in section 3.4.1.

2. The contextual descriptor of a test image is matched with the contextual descriptor of each of the clusters being created in the training phase. The distance computation to perform the matching is accomplished using *JSD*. Note that *JSD* here is used for another purpose. Previously in the training phase *JSD* is used to calculate the weight of FM or CM based on finding the target and background distributions. However, here *JSD* is now used to find the information distance between the contextual descriptor of the test image and that of each cluster. Note that the same *JSD* Eq. (3.4) is used. The only difference is that the target distribution is replaced by the contextual distribution of the test image and the background distribution is replaced with the contextual distribution of one of the clusters for which the matching is taking place.

3. The cluster having the lowest *JSD* value corresponds to the best match to the input image.

4. The precomputed weights of the FM and CM of the best match cluster are assigned to the input image.

5. The final *TD* saliency map using the selected weights is computed to produce the final result by following the procedure described in section 3.3.

By this procedure, a dynamic weighting of *BU* features is achieved based on the contextual content of the image. Figure 3.6 shows a sample of 43 training images being grouped into nine clusters by the above method. Furthermore, a sample test image is matched with the centroids of each cluster using *JSD*.

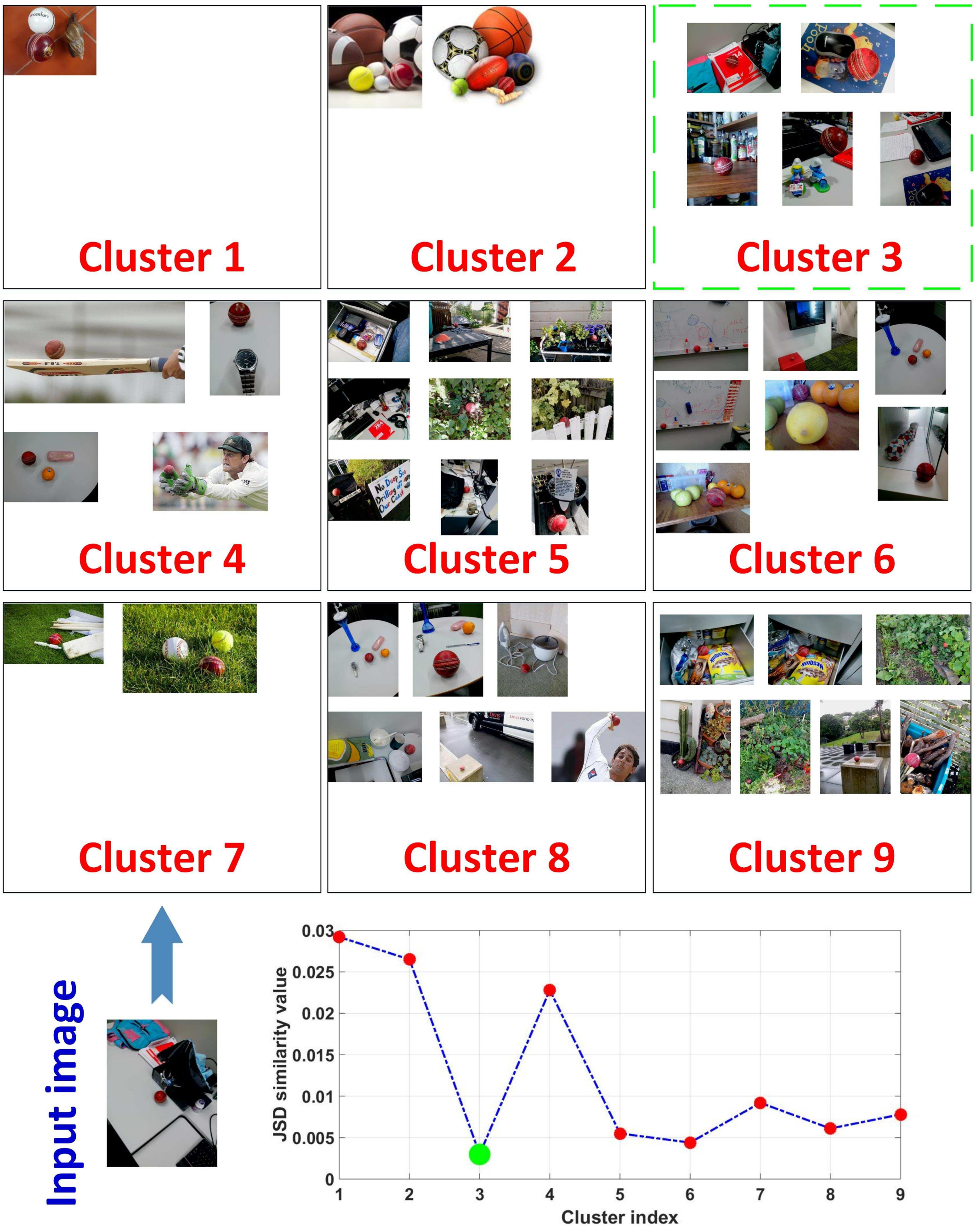


Figure 3.6: Contextual based image clustering illustration. 43 images are clustered into nine groups using the proposed *k*-mean contextual clustering technique. The similarity of images within a cluster using the contextual features can be seen in some clusters e.g. cluster four and five. An input test image is contextually matched to these clusters using *JSD* with cluster three being the best match (lowest *JSD* value (see the bottom plot).

3.5 Results and analysis

The experiments are divided into three parts. In the first part, the advantages of using the proposed features along with global *IDM* centre-surround mechanism are demonstrated over the traditional features and the centre-surround mechanism proposed in the *Itti* model. This evaluation is carried out for salient object detection. The results show that the detection accuracy of the salient objects on a benchmark dataset by the proposed features and the centre-surround mechanism outperform some state-of-the-art techniques in saliency detection including the *Itti* model [1].

The second part of the experiment shows that the proposed *JSD* weighting calculation produces higher precision, recall and *F*-measure values on a dataset that contains a cricket ball as the target object than that achieved by *JSD*. These three measures are indicative of the accuracy in detecting the target object (higher the better).

The last experiment shows the overall target object detection accuracy of the proposed *TDCoW* over four datasets and the importance of the contextual information in *BU* feature weighting for target object detection.

3.5.1 Experiment 1: bottom-up saliency performance

Two benchmark saliency datasets were used to demonstrate the effectiveness of the *BU* features used along with the global *IDM* approach of the proposed model for saliency detection. The first dataset called ASD or MSR-1000 is the leading benchmark dataset for saliency used by most researchers in this area [74]. It consists of 1000 images, each containing one salient object. The second dataset of 300 images is the SOD dataset, which is a collection of salient objects based on Berkeley Segmentation Dataset (BSD) [82]. According to the saliency survey paper [124], this is one of the most difficult and challenging datasets for salient object detection.

The precision-recall curve is used to evaluate the performance at different threshold values. The proposed method that utilizes different features and an *IDM* based centre-surround mechanism called *IDM* (multi-features) or simply *IDM* (Multi) is compared with 12 state-of-the-art fixation and saliency detection techniques as well as seven classical techniques. The *Itti* attention model is the baseline technique to compare against. The proposed saliency detection technique referred to as *IDM-PCA* [120] that utilizes the *PCA* for dimensionality reduction is also included for comparison. The rest of the techniques are abbreviated as follows:

- **Classical**: MSS [86], CA [76], Rare [125], SWD [94] and GBVS [91]
- **State-of-the-art**: BSL [126], CAU [127], RRWR [128], UFO [129], SIA(GC) [130], RC and HC [92], QCUT [131], IILP [33], HDCT [132], GMR [133] and FET [134]

The first set of methods (termed classical) are those methods that achieved good performance in saliency detection in the first two decades of the history of computational techniques for salient object detection. The state-of-the-art techniques are those that achieved the highest performance in salient object detection in recent years (refer to the survey papers on salient object detection techniques [27]).

Figure 3.7 shows the precision-recall curves obtained for both datasets. The obtained curves clearly show how *IDM* (Multi-features) outperforms all classical techniques for saliency detection. *IDM-PCA* has a good performance on both the datasets, however it uses a single colour feature with dimensionality reduction as opposed to *IDM* (Multi-features) which uses various effective and efficient features. In addition, the results also suggest the effectiveness of the features and the proposed *IDM* based centre-surround mechanism over those being used by the *Itti* model.

When comparing the proposed *BU* model with state-of-the-art techniques, it is obvious that the former has a very high precision value (higher than all other techniques) at low recall values (approximately below 0.4

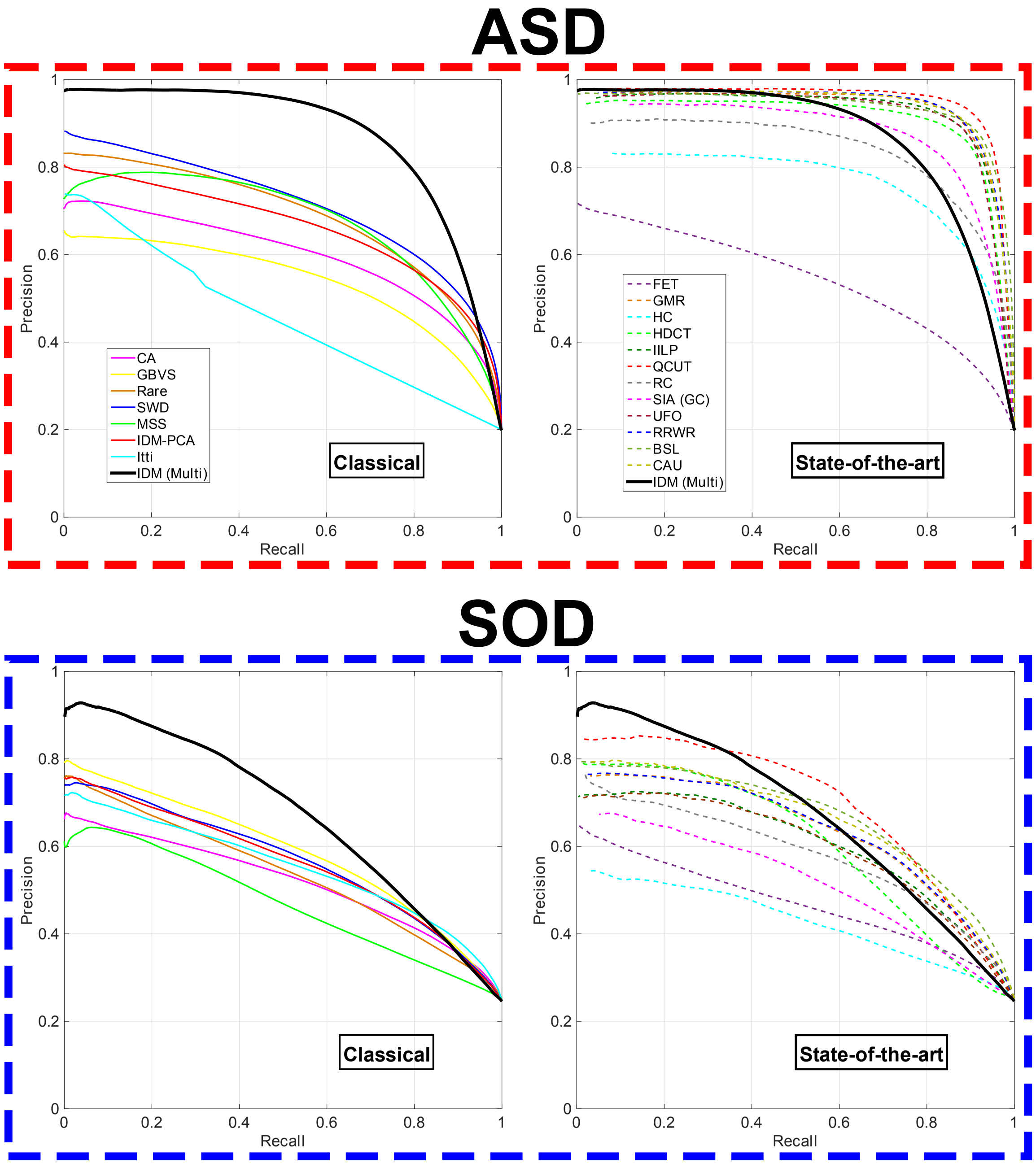


Figure 3.7: Precision-recall performance comparison of various classical and state-of-the-art *BU* techniques with the proposed *IDM* (Multi-features) model for bottom-up saliency detection on ASD and SOD datasets.

which correspond to low threshold) on the ASD dataset. The precision after this point degrades considerably. There are two reasons for this degradation in performance. Firstly, it is observed that the saliency maps generated by the proposed method typically highlight the salient object, only partially when the size of the salient region is large (a typical characteristic of the salient objects contained in this dataset). Secondly, most of the regions that are highlighted by the proposed technique exhibit low intensity values compared to other techniques. For this reason, when the threshold increases, the precision becomes low as the true positive value is small.

In the more difficult SOD dataset, it can be observed that the proposed technique comfortably outperforms all other state-of-the-art techniques for low recall values (approximately 0.5 and below). For high recall values, the degradation in performance is less obvious compared to the degradation occurring on the ASD dataset.

To visualize the maps generated by the proposed *BU* technique and other state-of-the-art techniques, Fig. 3.8 shows eight sample images taken from ASD and SOD datasets. The saliency maps generated by the all the techniques except for FET, HC and RC exhibit very high precision and low false negatives. Hence on this dataset, we can see from these sample images that almost all the techniques including the proposed *IDM (Multi-features)* were able to detect the salient object accurately if compared to the groundtruths (see second row of the figure). On the other hand, from the SOD dataset, four challenging images are selected in which they contain either more than one salient region to be detected (e.g., the last two images in this dataset) or the background and the salient region exhibit similar visual features (e.g., the first and the last image in this dataset).

In the first image from the SOD dataset, the groundtruth indicates that the salient region is the ladder in front of the rocky mountain. Note that the visual attribute of the background (mountain) and the salient region (ladder) are visually very similar. As a result, the pixel association and propagation based models such as *CAU*, *GMR*, *IILP* and *RRWR* assign

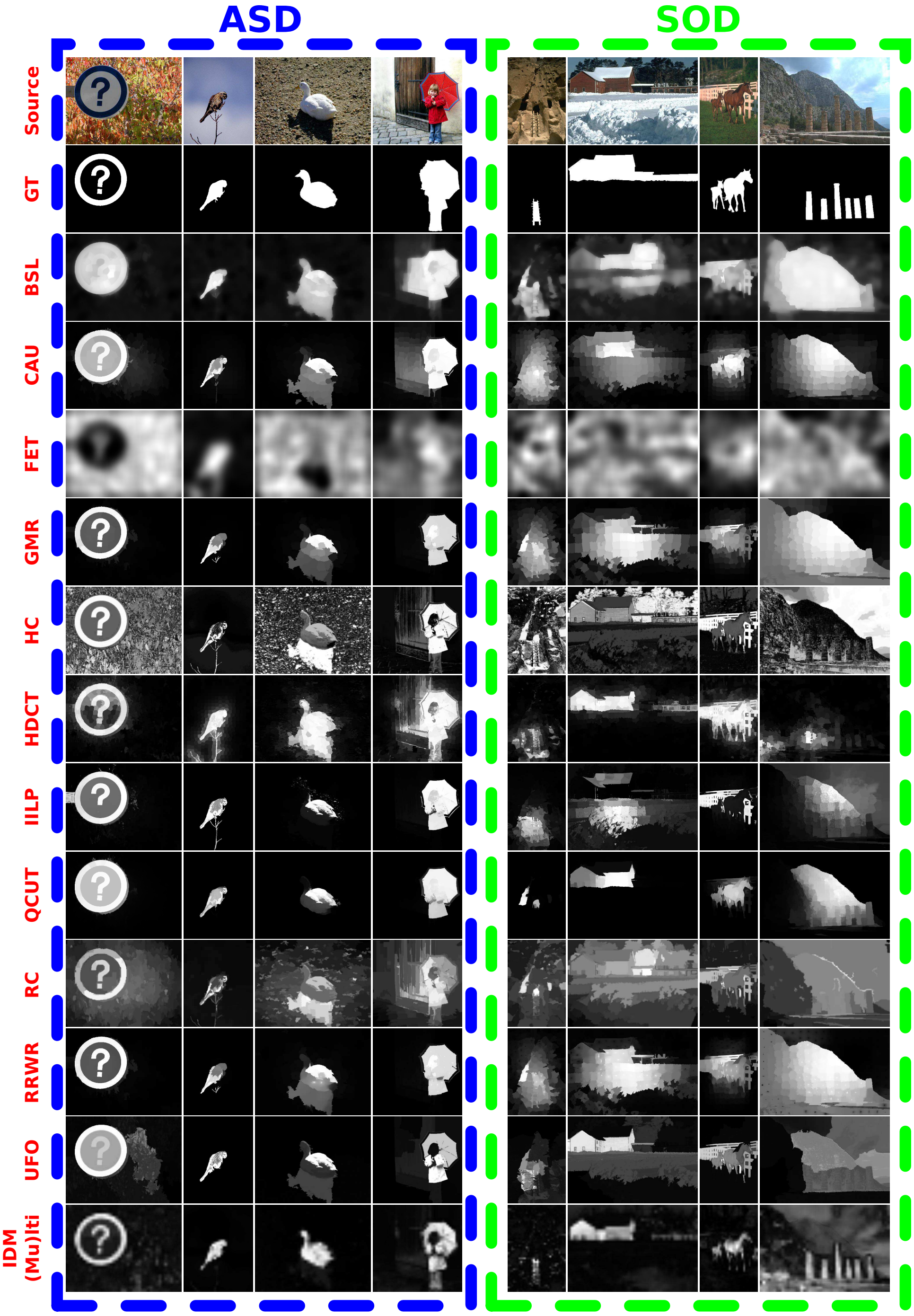


Figure 3.8: Qualitative comparison of the bottom-up saliency maps generated by various state-of-the-art techniques and the proposed *IDM* (Multi-features) on ASD and SOD datasets.

same pixels association values to the background and the salient region. Contrast based methods such as RC and HC also fail to produce satisfactory results as the contrast attributes between the two regions are similar. Even high-level features such as objectness in UFO was not able to separate the two regions. The best effort on this image was delivered by the proposed model (see the last row of the figure). Using an information theoretic approach to various low and mid-level features, the proposed model was able to highlight the salient region which is deemed to have certain irregularities or element of surprise from these features which in turn captured by the model. A similar visual analysis is applicable on rest of the images from this dataset.

The proposed *BU* saliency model highlights part of the salient object and mostly with low intensity, however, it retains the contour or the overall structure of the object. This is because one of the features used in this model is the *MSS* frequency content based on the method proposed in [86]. As mentioned earlier, the *MSS* technique has a very good segmentation performance on benchmark datasets [124].

Hence, to demonstrate the segmentation capability of the proposed *BU* model, the mean shift adaptive segmentation approach proposed in [74] is applied on the generated saliency maps to produce segmented binary saliency maps. The mean shift adaptive segmentation uses an adaptive threshold value for segmenting and binarizing the saliency map. Once this is done, the binarized map is compared with the groundtruth map and three parameters are computed, the precision, recall and F-measure (please refer to section 2.2.4.3 for the calculation equations). F-measure requires a β^2 parameter which is set to 0.3 (a common value found in saliency literature which gives more importance to precision than recall) [74].

From Fig. 3.9, we can clearly see that in both datasets, the proposed model has low recall values compared to other classical and state-of-the-art techniques. On the contrary, the proposed model exhibits high precision and in turn *F*-measure values. In the ASD dataset, the precision and *F*-measure values are comparable to most of the state-of-the-art techniques. In the SOD dataset, the proposed model has second largest precision and *F*-measure values after the QCUT technique. Although the recall values of the proposed model are low, the potential strength of the model lies in its ability to detect the salient object/s with a high precision.

Since the proposed *BU* model has a good performance for *BU* saliency detection, we would expect this proposed *BU* approach to perform well when modelling *TD* saliency as the *TD* weights are assigned directly to the features of this model.

3.5.2 Experiment 2: *JSD* weighting performance

In this experiment, the proposed *JSD* weighting calculation method is compared with the previously proposed *SNR* for *TD* feature weighting. For this experiment, the same setup being followed by the authors in [22] (i.e., same features and centre-surround mechanism based on the *Itti* model) is used to avoid any additional processing biases. The objective of this experiment is to show that *JSD* is a better option for feature weight calculation than *SNR*.

The *TD* weights are calculated with the two approaches once with the proposed *JSD* method and with the *SNR* method. A challenging dataset using cricket balls as the target is created for testing. The dataset consists of 400 images which are taken both indoors and outdoors with variation in the size of the target object, illumination, distracting objects and background. In addition, some internet images containing the cricket ball were included in the dataset. The reason for including internet images is to include natural images containing cricket balls (e.g. in well-known cricket grounds, with players, from different matches, etc.). The created dataset is split into two groups of 200 images each, where the first set contains images in which the target object is salient and the second in which the target is non-salient and distracted by other objects. They are referred to

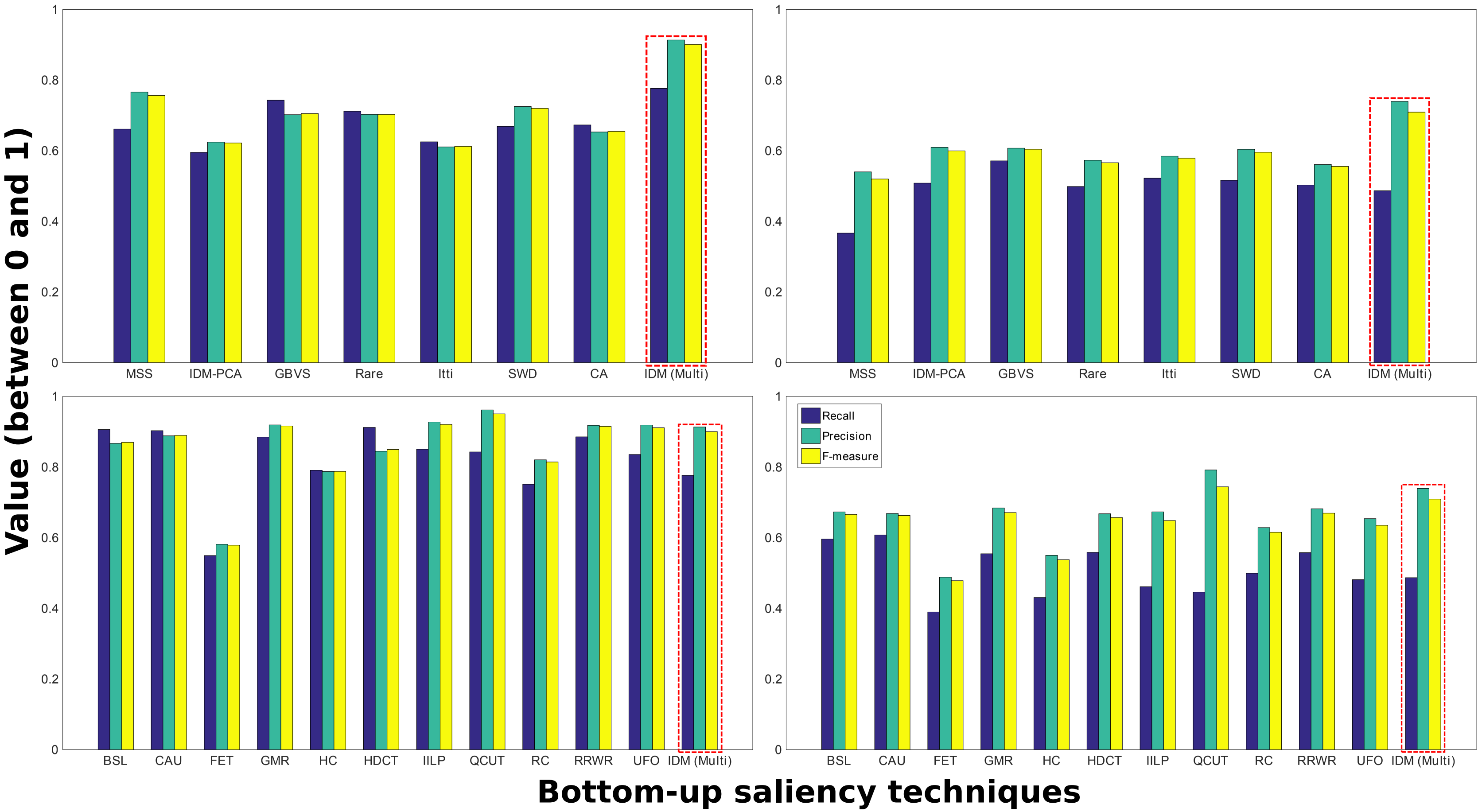


Figure 3.9: Segmentation evaluation of various models and the proposed model through precision, recall and *F*-measure values on ASD (the first column) and SOD (the second column) datasets. The first row represent the comparison with classical techniques and the second row for state-of-the-art techniques.

as salient and distractor datasets respectively.

For this experiment, 100 images (50 from each set) are selected randomly in total from both sets to construct a test to train instance ratio of $1 : 3$ (a common ratio followed in machine learning techniques). The experiment is repeated 10 times using a different random train/test image selection for each run. An average result is obtained in the form of precision-recall curves. As explained in section 3.4.2, the weights are calculated for the training images separately. These weights are then averaged to obtain a final single set of feature weights which are in turn universally applied on the testing images as there is no contextual or clustering involved in this experiment. The objective of this experiment is only to show that *JSD* is a better choice than *SNR* for weight calculation.

Figure 3.10 shows the average precision-recall curve for both the *JSD* and *SNR* based *TD* weighting along with the *BU* (i.e., no weighting) option. As expected, the highest precision value for the *BU* approach is very poor (approximately 30%). This is due to the fact that 50% of the test images are those in which the target is non-salient, and hence poorly detected by the *BU* approach. As it is evident, the *SNR* based *TD* weighting has improved the detection but only by approximately 10% of the maximum precision value. On the other hand, the *JSD* approach has a maximum precision value of 58% and clearly outperforms the *SNR* based approach. Note that the precision is still low due to the absence of contextual information when generating the feature weights. Figure 3.10 also shows a representative sample image and the obtained *TD* saliency maps from *SNR* and *JSD* respectively. As it is evident for this example, the *JSD* version has fewer false positive regions compared to the *SNR* alternative. This behavior was typical across the majority of the images in this dataset.

From the two experiments mentioned above, the effectiveness of the proposed features along with the global centre-surround mechanism and the *JSD* weight calculation procedure for both saliency and target detection is evident.

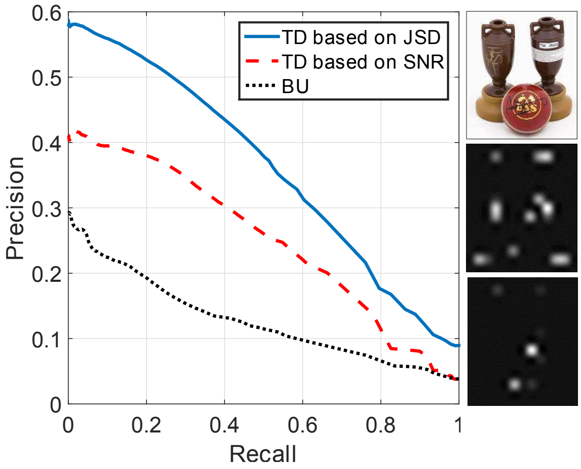


Figure 3.10: Comparison between *JSD* and *SNR* based weight calculation methods. The right column shows an example of a test image (top image) and the saliency map generated when applying *SNR* (middle image) and *JSD* (bottom image) as feature weight calculation procedure.

3.5.3 Experiment 3: *TDCoW* for target detection

In this experiment the efficacy of the proposed *TDCoW* and the importance of the contextual based clustering is explored. The model is tested on four datasets for target object detection. The first two are the same salient and distractor datasets discussed earlier for cricket ball target detection. The other two are selected from the Graz-02 dataset which is commonly used for object classification or recognition [135]. The Graz-02 dataset contains images with objects of high complexity and a high intra-class variability on highly cluttered backgrounds. There are three classes in this dataset, however, only two are considered for target detection i.e., bikes and persons as they are more difficult to be classified than the car class.

The images from each of the four datasets were split into equal halves, one for training and the other for testing. In addition, different cluster sizes were used. Figure 3.11 shows the average area under the curve (AUC) of the Receiver Operating Characteristic (ROC) achieved when varying the number of clusters in each dataset. It is clear from Fig. 3.11

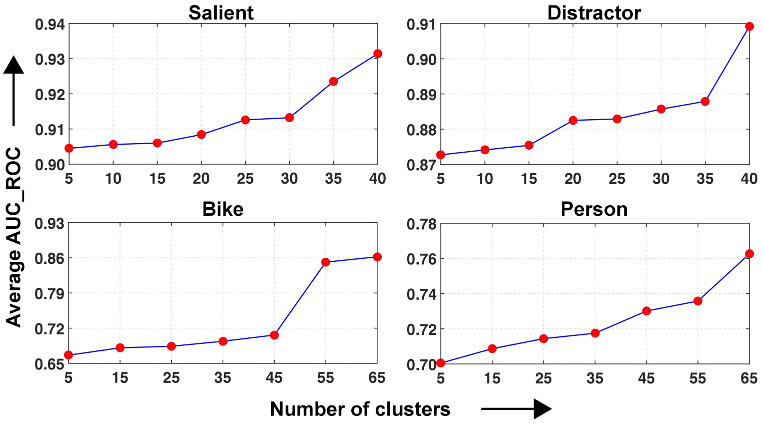


Figure 3.11: Cluster size variation affect on the accuracy of the proposed model.

that as the number of clusters is increased, a better AUC performance is achieved. The drawback of increasing the number of clusters is higher computational load in matching the contextual descriptor of the test image with that of the centroid contextual descriptors of the clusters. To be within a reasonable limit, empirically the number of clusters is chosen to be 30 for the salient and distractor datasets and 45 for the bike and person datasets.

3.5.3.1 Quantitative analysis

The model is tested by generating the *TD* saliency maps and finding the accuracy of detection using both precision-recall curves and the *F*-measure. A comparison is made between *TDCoW* and *TD* weighting without the contextual information or clustering. Figure 3.12 shows the results for the salient, distractor, bikes and persons datasets from left to right columns respectively where the top row is the precision-recall result and the bottom row is for the *F*-measure. The proposed *TDCoW* shows a better performance both in terms of precision-recall and *F*-measure curves across all four datasets. However, some observations and patterns need more elaboration.

Starting with the salient dataset, the *BU* (i.e., *IDM* (Multi-features) with

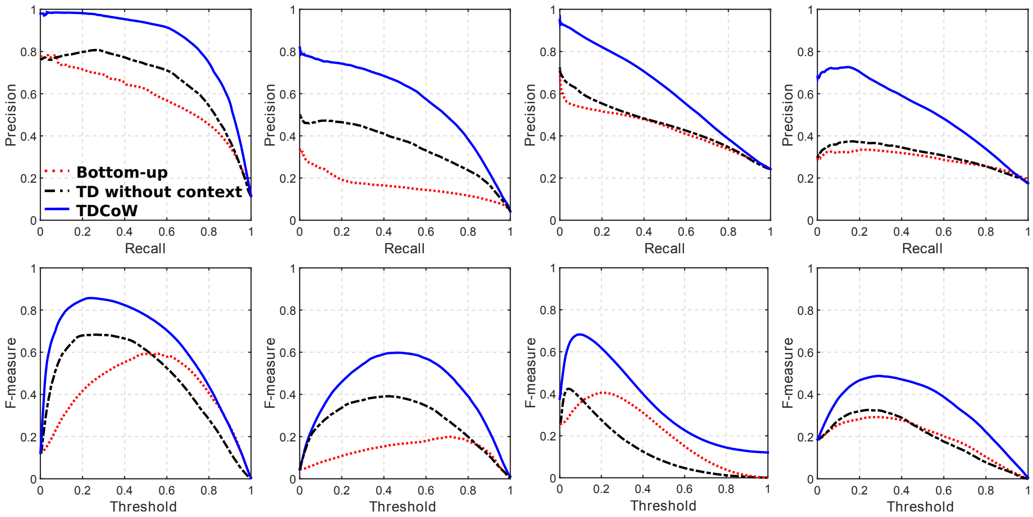


Figure 3.12: Precision-Recall and *F*-measure performance evaluation of *TDCoW* for the salient, distractor, bikes and persons dataset from left to right. The comparison is conducted with the *BU* model and the *TD* weighting without context. The top row is for Precision-Recall and the bottom one is for *F*-measure.

flat weighting on the features) has a reasonable performance in the accuracy of detecting the target (see the first column of Fig. 3.12). This is expected as we have seen the capability of the *IDM* (Multi-features) in section 3.5.1 in detecting salient objects, and this dataset has the target cricket ball being the most salient object in the image. When applying the weights without the use of context, the improvement is noticeable. For the distractor dataset, the *BU* approach gives a very poor performance as the target object is not salient and being distracted by other similar type of objects. Furthermore, weighting the features without the contextual knowledge results in a considerable improvement in the accuracy performance for this dataset. Finally, for the last two datasets, we can clearly see that weighting the features without the context has no affect in performance improvement over the *BU* approach. When using the proposed model (i.e., *TDCoW*) the performance is boosted considerably in all the four datasets.

Note that in the Graz-02 (bike) and (persons) datasets there is almost no difference in performance between the *BU* and the *TD* without context.

In fact, for some high threshold values, the *F*-measure values are higher for the *BU* than the *TD* without context. This suggests that for these two datasets in particular and in the absence of contextual information, the averaging over the examples is merely a random procedure and produces flat weights over all the features. This happens because the images in this dataset exhibit a very high inner-class contextual variability, which results in inaccurate weighting of the features when averaging the weights. This, in turn, leads to a degradation in target detection performance, as can be seen strikingly in the *F*-measure graph in third and fourth columns of Fig. 3.12 (compare the *BU* curves with the 'TD without context' curves).

From the three set of curves (precision-recall or *F*-measure) in all four dataset in Fig. 3.12, we can clearly see that learning weights through the incorporation of contextual information (*TDCoW* model) always outperforms both plain*BU* approach and feature weighting without context in terms of target detection accuracy performance.

Now to compare the proposed *TDCoW* model with existing state-of-the-art saliency techniques, again the PR and the *F*-measure curves are plotted to evaluate the performance of the proposed model. Furthermore, the model is compared with the model proposed by Judd et al. (LPH) [73]. This model is close to the proposed weighting model as it learns weights of various features from eye fixation data through an SVM classifier. Since the four datasets used in the experiments lack eye fixation groundtruth, it is not possible to train the weights over these datasets. Instead, from the segmented groundtruth region, a random sampling of points is performed to form an approximation of eye fixation data so that the model parameters can be learned from the training examples.

The precision-recall and *F*-measure curves are replotted in Fig. 3.13 from Fig. 3.12 for *TDCoW* to demonstrate the comparison with other state-of-the-art techniques. As before, the first row of Fig. 3.13 shows the precision-recall performance and *F*-measure values are plotted in the second row for all four datasets. On the first dataset (i.e., salient), it is evident

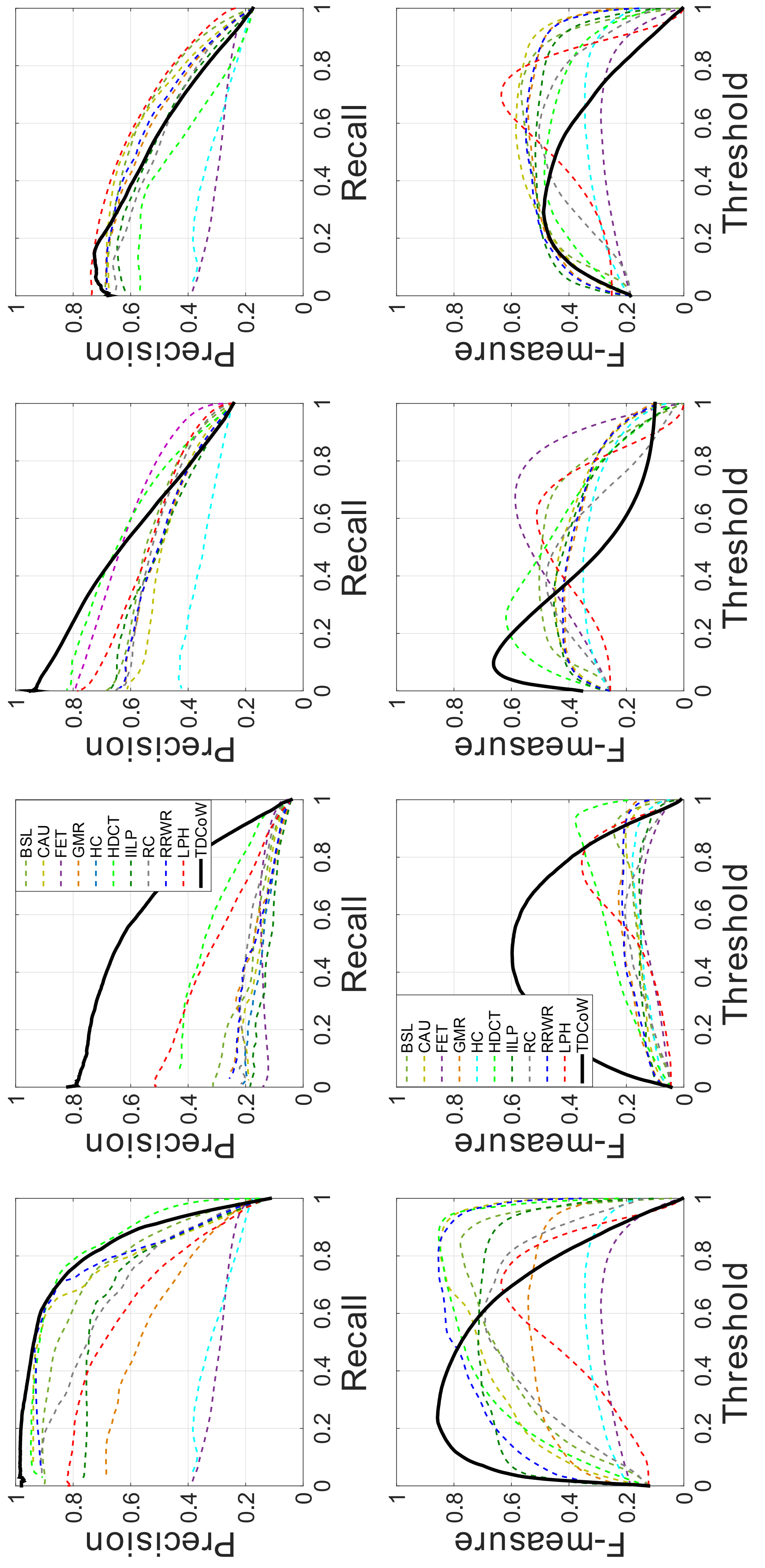


Figure 3.13: Precision-recall and *F*-measure comparison between *TDCoW* and other state-of-the-art-techniques for the salient, distractor, bikes and persons dataset from left to right respectively. The top row is for precision-recall and the bottom one is for *F*-measure.

that *TDCoW* has better performance than rest of the state-of-the-art techniques. Despite the cricket ball being salient in most of the images in this dataset, the *BU* state-of-the-art techniques could not perform as good as the proposed model.

The most striking performance of *TDCoW* can be seen for the distractor dataset. A huge performance difference between *TDCoW* and rest of the state-of-the-art techniques on this dataset confirms the capability of the proposed model in detecting the target object when it is not salient (see the first and second row of column two in Fig. 3.13). Since the distractor dataset contains distracting objects which are mostly salient, the poor performance of these techniques is reflected due to the fact that they falsely detected the most salient regions rather than the target object in majority of the examples on this dataset.

In the bike dataset, we can see similar performance by *TDCoW* to the one achieved on SOD dataset. Most of the images in this dataset contain the target object (i.e., bike) that are salient. As before, *TDCoW* has better performance on low threshold values than other techniques but degrades with increasing threshold. Similarly, the target object in the person dataset is also salient in most of the images. *TDCoW* has a moderate performance in this dataset. The best performance is achieved by LPH as the model uses high-level face and pedestrian detectors as features.

3.5.3.2 Visual analysis

Figure 3.14 shows representative examples of the saliency maps (shown by heatmaps) generated by various models including the proposed *TDCoW* model. The last three columns shows results produced when only using *BU* (i.e., the proposed *BU* IDM (Multi-features)), *TD* weighting but without context (*TD* (NC)), and finally the *TD* proposed model *TDCoW*. Three representative sample images are selected from each dataset.

As an example from the salient training image, the ball in the second image is salient, though it exhibits low contrast and is partially occluded by the grass. *TDCoW* was able to detect the target with high accuracy with a result comparable to HDCT and BSL. The rest of the techniques failed to detect the target precisely. In the distractor sample images, we can see that the proposed model outperforms the rest of the techniques in detecting the target with a small number of false positive regions.

The bike in Graz-02(bike) dataset was detected partially by *TDCoW* and not as a whole object. Despite the partial detection, the visual results compared to other techniques show very high target detection precision. As an example, in the third image, only the proposed model was able to detect the target with minimal false positive regions. On the other hand, the rest of the techniques detected the yellow object as it deemed more salient than the bike. In the person dataset, we can see a reasonable detection performance by the proposed model. For instance, in the last image, the object was detected but with far more false negative regions compared to more accurate results by other techniques.

When comparing the proposed model which is based on contextual *TD* weighting of features with the *BU* version (i.e., no weighting) and the *TD* weighting without context, we can clearly mark the visual improvement in locating the target object when using context (i.e., by *TDCoW* model) over the other two approaches (see the last three columns of Fig. 3.14). In the majority of these sample images, we clearly see that pure *BU* has poor performance in detecting the targets, particularly in the distractor and bike datasets. Little improvement is achieved when performing *TD* weighting of features over all the training examples without the inclusion of context. Ultimately, upon incorporating the context, the detection performance is obvious. In some situations, for instance (the improvement in second and third image of the distraction and person datasets respectively), the incorporation of context does not have a significant improvement over the *TD* weighting without context. In other occasions, a noticeable improvement is achieved when using the context to modify the *TD* weighting either by increasing the precision in detecting the target (e.g., the first image int the

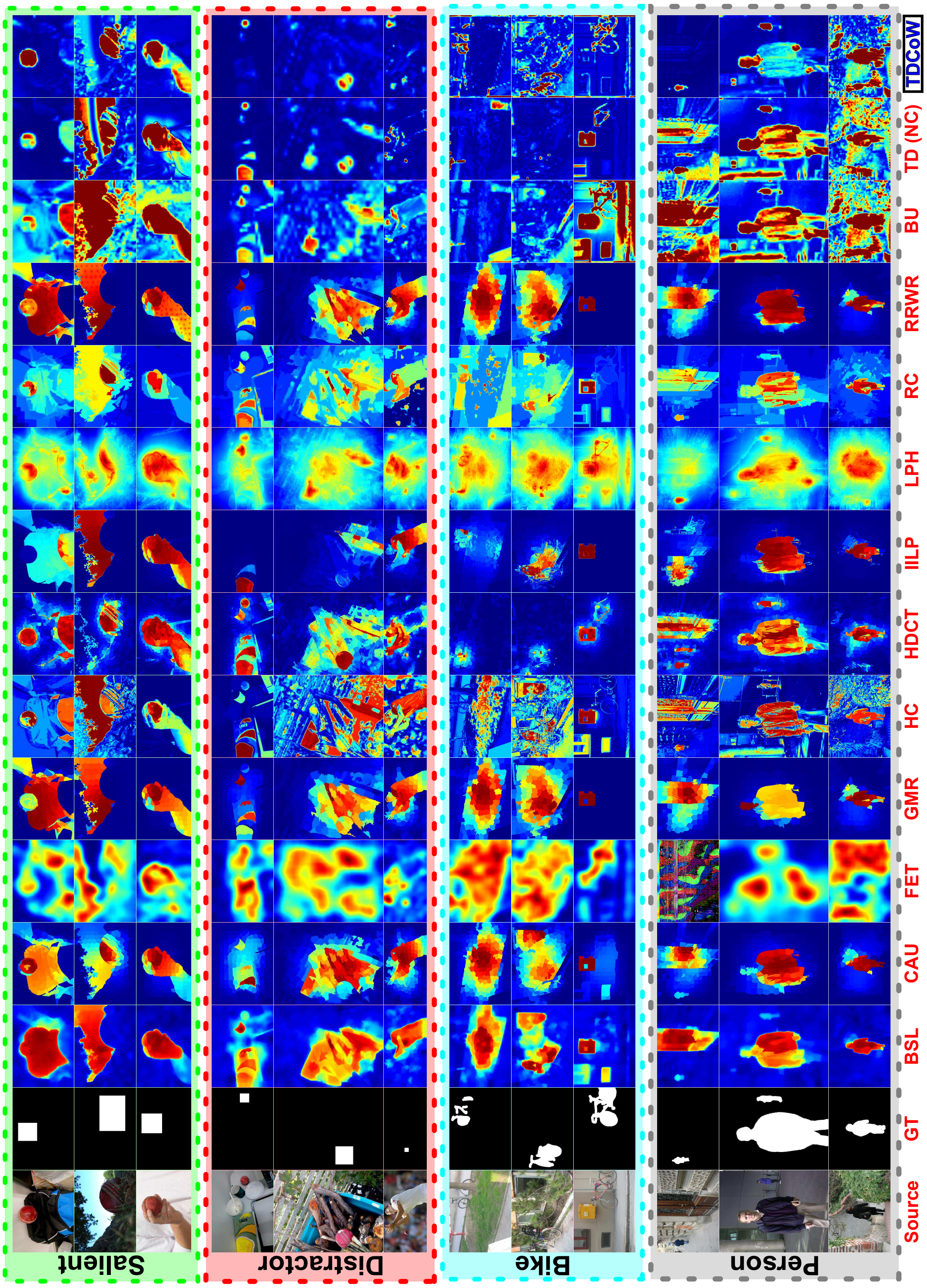


Figure 3.14: Qualitative sample saliency maps produced by the proposed *TDCoW* and other state-of-the-art models.

bikes dataset) or reducing the number of false positive regions (e.g., the first image in the distractor dataset).

3.5.4 Limitations of *TDCoW*

Although the model utilizes the image context for a better feature weight assignment, the knowledge of the target features is missing. Target features remain an important factor for more accurate weighting of features. In addition, other sources of visual guidance such as task encoding, long-term memory of previously attended objects and location priors can be used for a finer tuning of the feature weights. However, such high-level information of the scene requires more processing time. Hence, a trade-off exists between the choice of information sources that could assist in target object detection and the efficiency of the system.

3.6 Chapter summary

Modelling top-down saliency by appropriately weighting the bottom-up features for target detection is a non-trivial research topic in active vision. The major challenge in top-down saliency modelling is how to dynamically assign weights to the bottom-up features. Most of the existing techniques do not consider high-level information within an image when weighting the features. As a result, the learned weights from example images only work well when the test images are similar in their context or background to the training images.

To overcome this problem, the proposed Top-down Contextual Weighting (*TDCoW*) model learns contextual structures from the training images and applies them to the test images in order to dynamically assign weights to the features. A clustering approach is used for appropriately cluster similar types of images in the training phase. The cluster weight is assigned to the test image that corresponds to the best match to the test image context. Upon doing the context based weighting, the proposed model (*TDCoW*) outperforms many state-of-the-art *BU* techniques as well as top-down weighting without context in detecting the target object in four challenging datasets. The results obtained by the proposed model in all four datasets show a considerable target detection performance improvement through precision-recall and *F*-measure values over feature weighting without context.

Hence, the major contribution of this chapter is to highlight the importance of contextual information for top-down saliency modelling by feature weighting for target detection. Furthermore, in order to achieve the mail goal of this chapter, the following sub-contribution were made:

1. A new proposed bottom-up salient object detection technique called information divergence of multi-features (*IDM*-(multi-features)) was proposed to produce saliency maps that better detected salient objects than many state-of-the-art bottom-up saliency techniques in benchmark datasets.

2. A new information theoretic based technique for computing the feature weights. The proposed approach based on Jensen Shannon divergence *JSD* produces bounded and more accurate weights for modelling top-down saliency in detecting target objects than those produced by conventional signal to noise (SNR) method proposed in [22].

3. A new descriptor is proposed based on the feature distributions that accurately extracts the gist content of an image.

As highlighted in section 3.5.4, contextual information is not the only source of tuning the attentional feature weights for modelling top-down saliency. In the next chapter, we incorporate a target information to the contextual model through two different mechanisms without using any high-level information of the target object.

Chapter 4

Modelling Top-Down Visual Attention Through Target and Contextual Information

4.1 Chapter introduction and motivations

As demonstrated in the previous chapter, weighting low-level attention features to accommodate target object visual search can be improved through the use of contextual information contained in an image. However, contextual information is not the only possible source of guidance when performing target object detection [30]. Target object features play an essential role in guiding the attention towards the object of interest. This is evident in many passive computer vision techniques [136], active vision techniques [34] and computational visual attention models [8].

The biological phenomenon of visual attention suggests that when searching for a target object, a lot of information of the scene such as spatial distribution, contextual information, bottom-up attention and prior knowledge of the target features act simultaneously in guiding the search towards the target [4, 37]. It has been speculated that when a candidate region of interest is sampled, a recognition process is triggered to identify the region as target or non-target [37]. This suggests that two levels of the target feature involvement exist when performing a top-down visual search; one during top-down sampling of a visual scene and the other when performing the final recognition of the sampled region.

Modeling TD attention by utilizing both guidance source (i.e., contextual information and target features) could result in regions of interest that are more likely to contain the target object than that acquired from any individual sources. Previous models have only considered target features either during the attention phase or at the recognition stage when modelling top-down saliency [71,99]. As a more comprehensive top-down attention system, in this chapter we combine contextual information and target feature at the attention phase (from now it is referred to as *attention target features (ATF)* and again use target features at the recognition phase (referred to as *recognition target features (RTF)* to model TD saliency.

Figure 4.1 shows the three modules in this work and how they interact with each other. Both the contextual and ATF modules produce maps representing regions of interest to be attended to. By contrast, the RTF module processes these maps by inspecting each region to find the likelihood of it containing the target object. While it is clear from the figure that the target features used by RTF and ATF share the same type of features, the ways they utilize them are different.

At this stage it is important to distinguish between the purposes of the contextual and ATF modules and the RTF module for recognition. The first two modules are the key elements of the whole model because they are responsible for generating saliency maps that are likely to contain the target object. The RTF module only inspects the salient regions to confirm the detection process. Hence, the RTF is a generalized module that can be applied to any saliency map which is fed to.

While the main contribution of the chapter is the integration of the contextual and the target features to maximize detection accuracy, the recognition module can be considered as a secondary contribution of this chap-

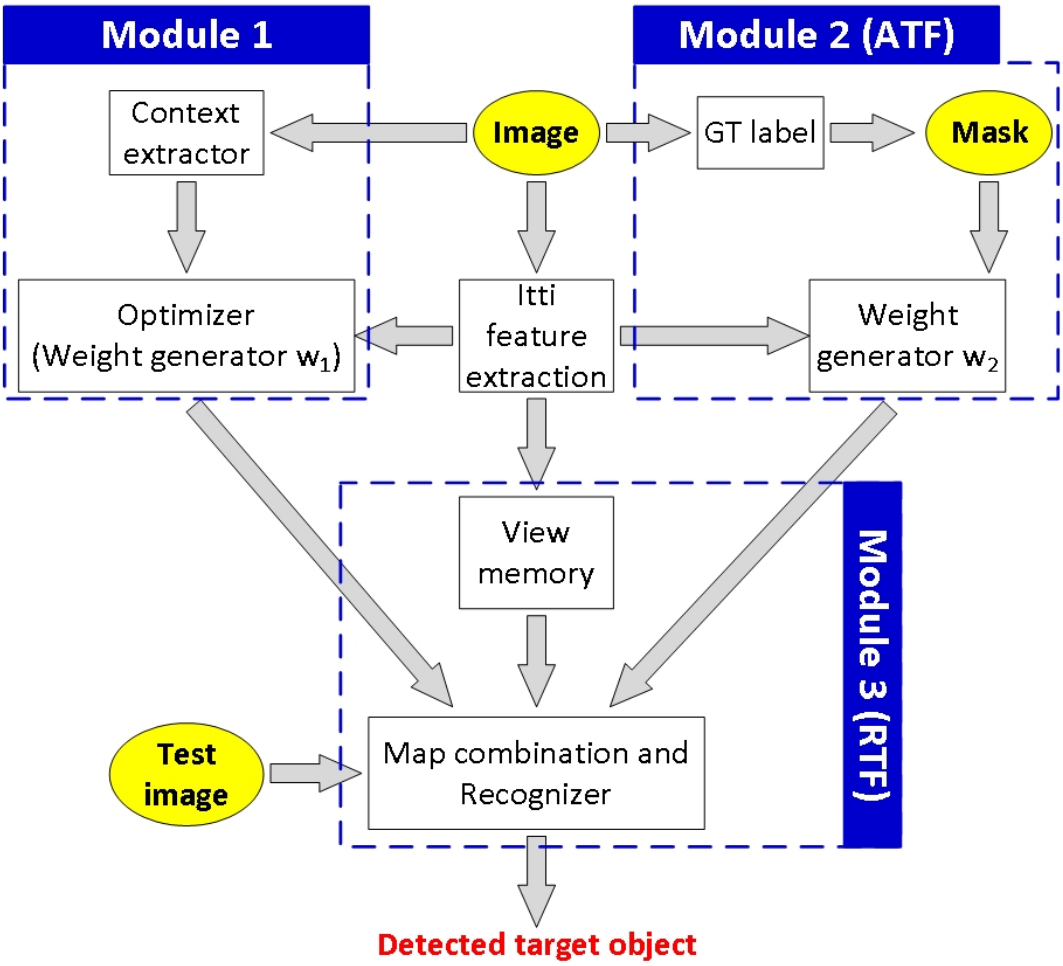


Figure 4.1: A block diagram of the proposed framework for modeling top-down attention feature weighting for target object detection. The model consists of three modules, two for generating saliency maps and one is for the recognizing the target object.

ter because it utilizes the same basic information used in the ATF module effectively and efficiently without opting for complex computer vision recognizers. Similar to chapter 3, in this chapter, there are various implementation details that either directly or indirectly contribute towards building the final model. However, the major contribution of this chapter is the incorporation of the target information through target controlled weighting mechanism (i.e., the ATF module). This contribution beside being a proof of concept, provides a way to incorporate target information into the attention system using low level feature without opting for high level target specific features. To our knowledge, such an approach has not been considered previously in visual attention based object detection.

4.1.1 Chapter objectives and overview

The objective of this chapter is to improve the target detection accuracy of the existing top-down feature weighting attention model through the incorporation of both contextual and target information into the system. In order to achieve this goal, three modules of the overall system are introduced and the following tasks are considered,

- To learn low-level weights for the features (indicated as $\mathbf{w}_1$) that maximize the detection accuracy of the target object through feature space optimization. For this purpose, the state-of-the-art particle swarm optimization (PSO) global search technique is used. The optimization of weights is performed over individual images. Thus each image will have its own optimized set of weight vectors.

- Compute the contextual information of an image. Three different contextual descriptors are used to explore the ability of these descriptors to summarize the holistic information within an image, which in turn would be reflected in the quality of feature weighting. This particular objective extends the previous work on contextual descriptors presented in the chapter 3 by using a set of more effective descriptors.

- To learn a model that accurately exploits patterns between the $\mathbf{w}_1$ weight vectors and the contextual descriptors from a set of training images. Later, this model will be used to predict $\mathbf{w}_1$ from the contextual descriptor of a novel image.

- To learn a separate set of weight vectors referred to as target object weight vectors ($\mathbf{w}_2$). The weight vector is based on extracting low-level features from the target object to be searched for. This weight differs from $\mathbf{w}_1$ as it maximizes the detection accuracy by considering only target related features. The $\mathbf{w}_1$ weight vector does not consider any target related information. Another difference is that $\mathbf{w}_2$ represents a single weight vector that is applied on all the images in a dataset whereas by contrast, $\mathbf{w}_1$ is the set of weight vectors that is learned for each image separately.

- Formulate a mechanism to extract candidate regions from the saliency map followed by applying recognition to determine the likelihood that the region contains a target object.

Briefly, the below procedures are followed to build the final model for integrating target and contextual information for object detection,

1. Split the data into training and testing sets.

2. For each image from the training set, learn an optimized feature weight vector using PSO (for details refer to section 4.2.1.1).

3. Compute the contextual information of each image using one of the three contextual extraction techniques provided in section 4.2.1.2.

4. Using a hetero-associative single layer neural network, learn an association model between the weights from step 2 and the contextual descriptors obtained from step 3.

5. For all the training images collectively compute a single set of weights (target weights) from the ATF module by extracting low-level features (for details refer to section 4.2.2). Note that this set of weights is different from the weights in step 2 as the latter represent a single set of weights for the target object in all the training images.

6. From the same low-level features extracted in from previous step, learn a Naive Bayes classifier for binary classification (i.e., target and non-target) (for details refer to section 4.2.4). This is the optional RTF module which represents the second level of target detection after ATF.

7. For a novel image from the testing dataset, compute its contextual descriptor.

8. Apply the hetero-associative model on the descriptor to yield the first set of weights (we refer to it as contextual based weights).

9. Compute the saliency map through *Itti* model using the weights obtained from the previous step. We refer to this saliency map as contextual based saliency map.

10. Using the target based weights obtained in step 5, generate another saliency map through the *Itti* model. We refer to this saliency map as target based saliency map.

11. Combine both saliency maps to form the combined saliency map using one of the combination operations given in section 4.2.3.

12. Segment the obtained combined map using a mean-shift segmentation technique.

13. For each salient region in the segmented map, apply the Naive Bayes classifier to classify these regions to either belonging to the target or non-target object.

14. The final output of the model is a saliency map containing regions that most likely contain the target object to be detected.

4.2 Model structure

The proposed model is based on generating various saliency maps that are computed through different feature weighting mechanisms. Each set of learned weights tunes the attention features differently, yielding maps that highlight a variety of potential regions that are likely to contain a target object. The basic Itti features are used to compute the saliency maps [1]. Referring to Fig. 2.5, these features are red/green (*R/G*) and blue/yellow (*B/Y*) representing opponent colour contrasts, intensity (*I*) and four orientation features computed at 0°, 45°, 90° and 135° using Gabor filters. Furthermore, the same centre-surround and normalization operations are used as in the original *Itti* model.

4.2.1 Module 1: contextual weighting

The purpose of this module is to learn a set of weights for the above mentioned features that maximizes the detection accuracy of the target object. Similar to the weight vector used for TDCoW in section 3.2, a weight is associated with each feature map. In addition to the feature weights, weights are associated to conspicuity maps as well. Thus, with the basic Itti features, we have a total of 10 weights, three for each conspicuity map and the remainder for the feature maps in the corresponding feature sub-channels. So the set of weights to be learned represents a feature vector $\mathbf{w}_1$ represented as:

$$\mathbf{w}_1 = \left[w_{R/G}, w_{B/Y}, w_I, w_{0^o}, w_{45^o}, w_{90^o}, w_{135^o}, w^C, w^I, w^O \right] \tag{4.1}$$

where the last three weights are the conspicuity map weights for colour, intensity and orientation respectively.

As shown in Fig. 4.2, for each image in a dataset of K images, we learn an optimum feature weight vector $\mathbf{w}_1^i$ separately through optimization. The optimization block takes two inputs, the image i for which we want to learn the weights and its groundtruth mask.

The optimizer selects values for $\mathbf{w}_1^i$ weight vector in such a way that when the weights are used on the feature and conspicuity maps to yield a saliency map, it best matches the groundtruth map. This agreement or detection accuracy is measured through *F*-measure score. Furthermore, The optimization takes place simultaneously for the whole weight vector. This is different from previous approaches for optimizing saliency features [23], where the same type of sub-channels features (not the conspicuity maps) are assumed to be independent and hence optimized separately (e.g., the set of weights for colour are optimized separately than those associated with orientation for instance). Because such optimization is performed on separate sub-channel features, and because some of the existing valuable dependency and interaction between these sub-channel features are not considered, the resultant weights would be far from being optimized and would affect the detection performance. For this reason, in the proposed optimization approach, all the sub-channel features are optimized simultaneously. Note that at this stage, we are not concerned with splitting the dataset into training and testing sets, instead, weights are pre-learned for each image to ease computation in the successive training phase for learning contextual association.

The selection of the optimization technique is a design issue that depends on the requirement. For instance, the authors in [111] chose constrained non-linear programming as it is fast to implement. However, this technique tends to provide solutions that are non-optimal. In the case of optimizing the weights for maximizing detection accuracy, the requirement for the optimizer is to provide near optimal solutions (in this case weights). A non-optimal weight will not only affect the contextual weighting module but also affect the overall performance of the model's target

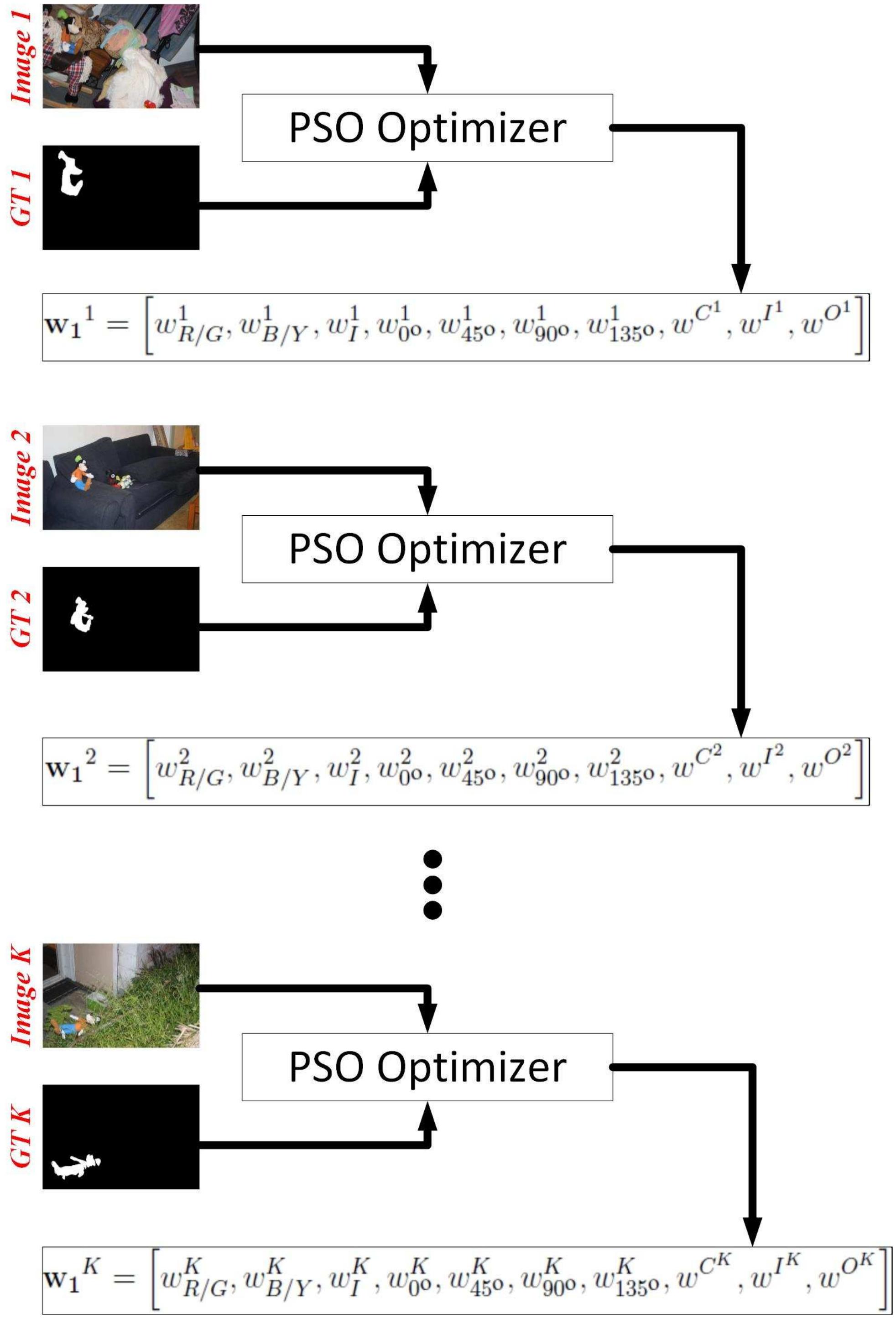


Image 1
GT 1
PSO Optimizer
$\mathbf{w_1}^1 = \left[w^1_{R/G}, w^1_{B/Y}, w^1_I, w^1_{0^o}, w^1_{45^o}, w^1_{90^o}, w^1_{135^o}, w^{C^1}, w^{I^1}, w^{O^1}\right]$
Image 2
GT 2
PSO Optimizer
$\mathbf{w_1}^2 = \left[w^2_{R/G}, w^2_{B/Y}, w^2_I, w^2_{0^o}, w^2_{45^o}, w^2_{90^o}, w^2_{135^o}, w^{C^2}, w^{I^2}, w^{O^2}\right]$
Image K
GT K
PSO Optimizer
$\mathbf{w_1}^K = \left[w^K_{R/G}, w^K_{B/Y}, w^K_I, w^K_{0^o}, w^K_{45^o}, w^K_{90^o}, w^K_{135^o}, w^{C^K}, w^{I^K}, w^{O^K}\right]$

Figure 4.2: The procedure for generating $\mathbf{w}_1$ weight vector for all the images in a dataset. The optimization is performed independently for each image.

detection accuracy. Furthermore, because our proposed model can be expanded to any number of features, the optimizer should be capable of searching very large spaces of possible solutions.

There is a family of optimizers that can effectively search complex and high dimensional feature space to yield solutions that coincide more often to global optima [137]. The particle swarm optimization (PSO) is one such technique that falls into this category and fulfills the above mentioned requirements. Since the search space structure for the current problem is complex and not known, we can not make any assumptions about it (which is sometimes required by optimizers). PSO fits such problems as it does not require any prior assumption of the search space as it is a population based [138, 139]. In addition, PSO is easy to implement and has a fast convergence rate.

4.2.1.1 Particle swarm optimization for feature weighting

Particle swarm optimization (PSO) is an evolutionary computation technique that uses particles or agents to describe different possible solutions in a search space [138]. Each particle k is described by three entities, its current position in the search space denoted as $\mathbf{x}_k$, its velocity denoted by $\mathbf{v}_k$ and the its fitness value f_k. Usually, these entities are described in a vector of length D where D is the dimensionality of the search space. The fitness value of a particle is evaluated using some fitness or objective function.

The basic principle of PSO is that after every iteration, the particles update their velocities and positions in that the overall flow is towards a particle/s having the best overall previous fitness value. That particle is called the global best (G_{best}) and its position is denoted as g. In addition, each particle records its personal best position (P_{best}) denoted as $\mathbf{p}_k$ (i.e, the position having the best fitness value). Note that g and $\mathbf{p}_k$ are also vectors of dimension D.

At each iteration t, the particles update their velocity and positions according to the following equations for the standard PSO [140]:

$$\mathbf{v}_{kl}^{t} = w\mathbf{v}_{kl}^{t-1} + c_1 r_1(\mathbf{p}_{kl} - \mathbf{x}_{kl}^{t-1}) + c_2 r_2(\mathbf{g}_l - \mathbf{x}_{kl}^{t-1}) \tag{4.2}$$

$$\mathbf{x}_{kl}^{t} = \mathbf{x}_{kl}^{t-1} + \mathbf{v}_{kl}^{t} \tag{4.3}$$

where c_1, c_2 and w are various scalar parameters for performance tuning of PSO. The random numbers r_1 and r_2 are generated from a uniform distribution between zero and one. The variable l is used to index the elements of the vectors such that $l = 1, 2, \ldots, D$. The update policy can be either global (i.e., each particle in the search space acts as an informer to other particles) or local (i.e., particles belonging to the same group update each other only in a ring topology) [140]. This formation of such structural grouping is referred to as the neighborhood topology. A global mesh topology has the disadvantage of providing a local optimum solution whereas local ring topologies are used to overcome this problem [141,142].

In the context of the weighting problem, each particle is represented by a weight vector $\mathbf{w}_1$ and hence, the search space is $D = 10$ dimensional such that the weights have bounded values between zero and one. In order to evaluate the fitness value of a particle, the detection accuracy is computed as follows,

- For a given weight vector that represents a particle position, the saliency maps is computed after weighting the feature and conspicuity maps with the corresponding weights from the weight vector.
- The saliency maps is compared with the groundtruth map of the image in consideration. Any saliency detection accuracy measurement can be used from section 2.2.4.3. Since we would like to balance between precision and recall, the *F*-measure score is used. The *F*-measure score is computed by first thresholding the saliency map using a single adaptive thresholding value and then computing the *F*-measure score using Eq. (2.2) between the thresholded saliency map

and the groundtruth map. The F-measure score represents the final fitness value of the particle. An example of computing the fitness value of a particle is demonstrated in Fig. 4.3.

As a result, the objective function of optimization is to maximize the F-measure value or equivalently to minimize $1 - \mathbf{f}_k$. The optimization process is performed over multiple runs. Each run is a complete optimization that is performed over a number of iterations. In each run, the seed (and the population initialization) is different which results in diversity in solutions. Each run produces a winner solution (i.e., a G_{best} particle having the best fitness value). The final set of optimum weights is selected having the best fitness value over all runs. We have chosen the best solution instead of taking average over runs as the latter may result in weight variation on certain features, particularly when the search space in not convex. As a result, if the weights vary on certain features vary considerably from the best solution weight vector, the accuracy of the generated saliency map will be affected.

Note that the whole process described for acquiring an optimum weight vector is performed on a single image and it is repeated for the all the images in the dataset separately to obtain an optimum weight matrix given as:

$$\mathbf{W}_1^{opt} = \left[\mathbf{w}_1^1, \mathbf{w}_1^2, \ldots, \mathbf{w}_1^K\right]^T \tag{4.4}$$

As will be discussed in section 4.2.1.3, this weight matrix is further used in the contextual association step to learn a pattern between the context of the image and these weight vectors.

4.2.1.2 Contextual information extraction

The weight vector $\mathbf{w}_1^i$ of an image i represents the optimum set of weights that maximizes the detection accuracy for that particular image. Hence, applying the same weight to a different image will cause a performance change (mostly degradation), unless the structure, content and visual at-

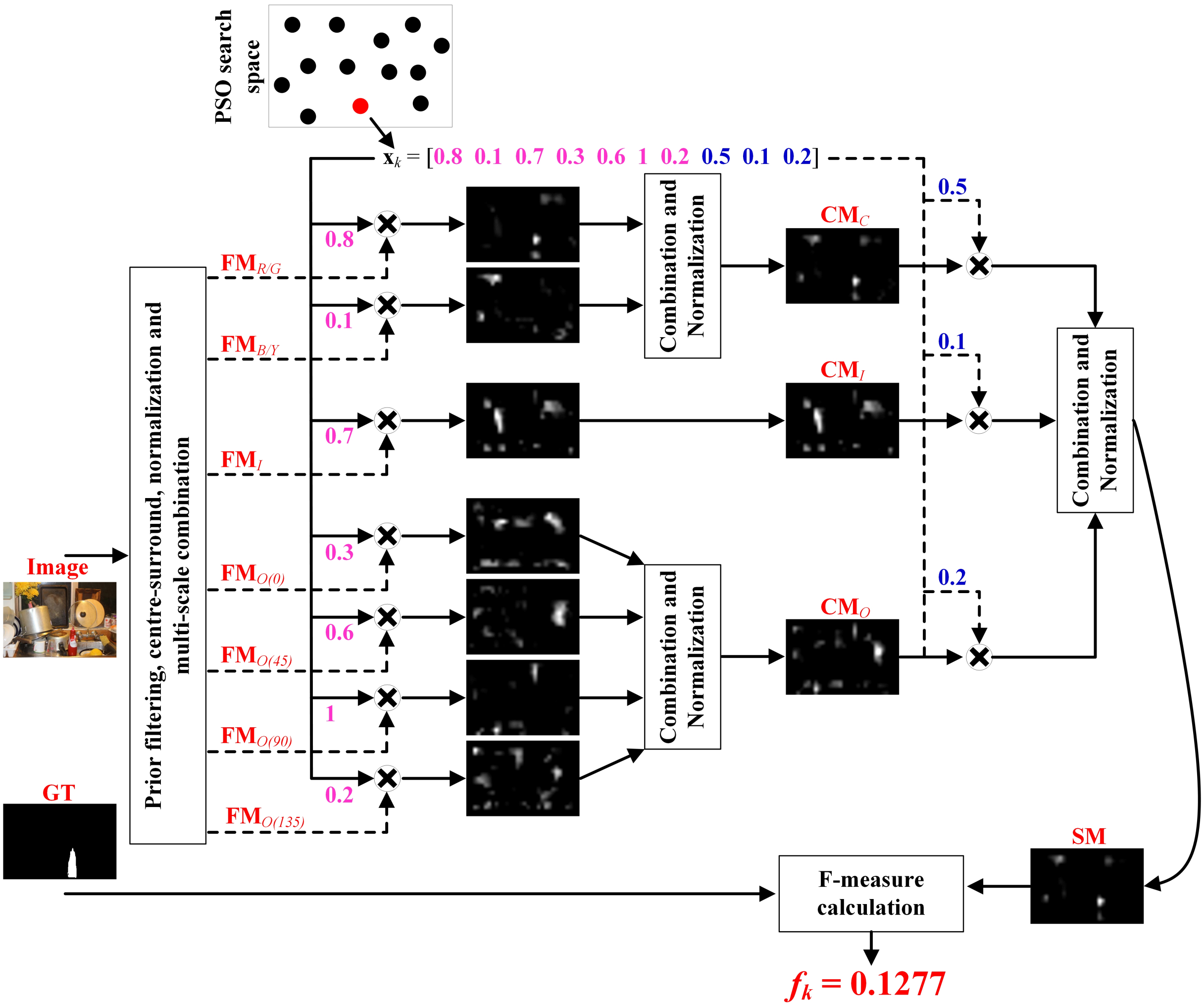


Figure 4.3: The procedure for computing fitness value of a particle through an example. A particle (shown as a red circle) is encoded as a weight vector. The weight vector is used to weight the corresponding feature and conspicuity maps to yield a top-down saliency map. The generated map is compared with the groundtruth map through an adaptive threshold value to yield the final *F*-measure value. This value represents the fitness function of that particle.

tributes of the image remain the same. This can be demonstrated by inspecting some images shown in Fig. 4.4. By viewing the four images, we can roughly categorize the images by their visual similarity. For instance (from left to right of the first row of the figure), the first and second image exhibit similar background as are the third and fourth. For each image, a weight vector $\mathbf{w}_1$ is learned through the optimization procedure described above.

For instance, when the saliency map of the third image was generated through its optimized learned weight vector (i.e., $\mathbf{w}_1^3$), a detection accuracy value of 0.422 was achieved (as measured through F-measure segmented evaluation procedure described in section 2.2.4.3). When the saliency map of the same image was produced by the optimized weight vectors obtained for the first and second images, a poor detection performance was resulted (F-measure values of 0.068 and 0.107 respectively). When $\mathbf{w}_1^4$ was used, a performance of 0.336 was achieved. In fact, by looking at the four saliency maps, those produced by the weight vectors of the first two images are similar in visual characteristics whereas as the one produced by $\mathbf{w}_1^4$ is highly similar to the one achieved by the optimum weight vector for this image (i.e., $\mathbf{w}_1^1$).

The example suggests that when the visual attributes of the images are similar, then the weights learned for one image can be applied to another to achieve similar detection performance. For a novel image, because there is no groundtruth map indicating the location of the target object, its feature weights are assigned with the help of those examples that exhibit high similarity in contextual information to that of the novel image. For this reason, and motivated by this observation, the contextual information needs to be extracted from the image that would give a holistic gist representation of its contents and serves as a content visual descriptor for that particular image.

Note that the concept of weighting the features based on the contextual information has already been introduced in chapter 3 when propos-

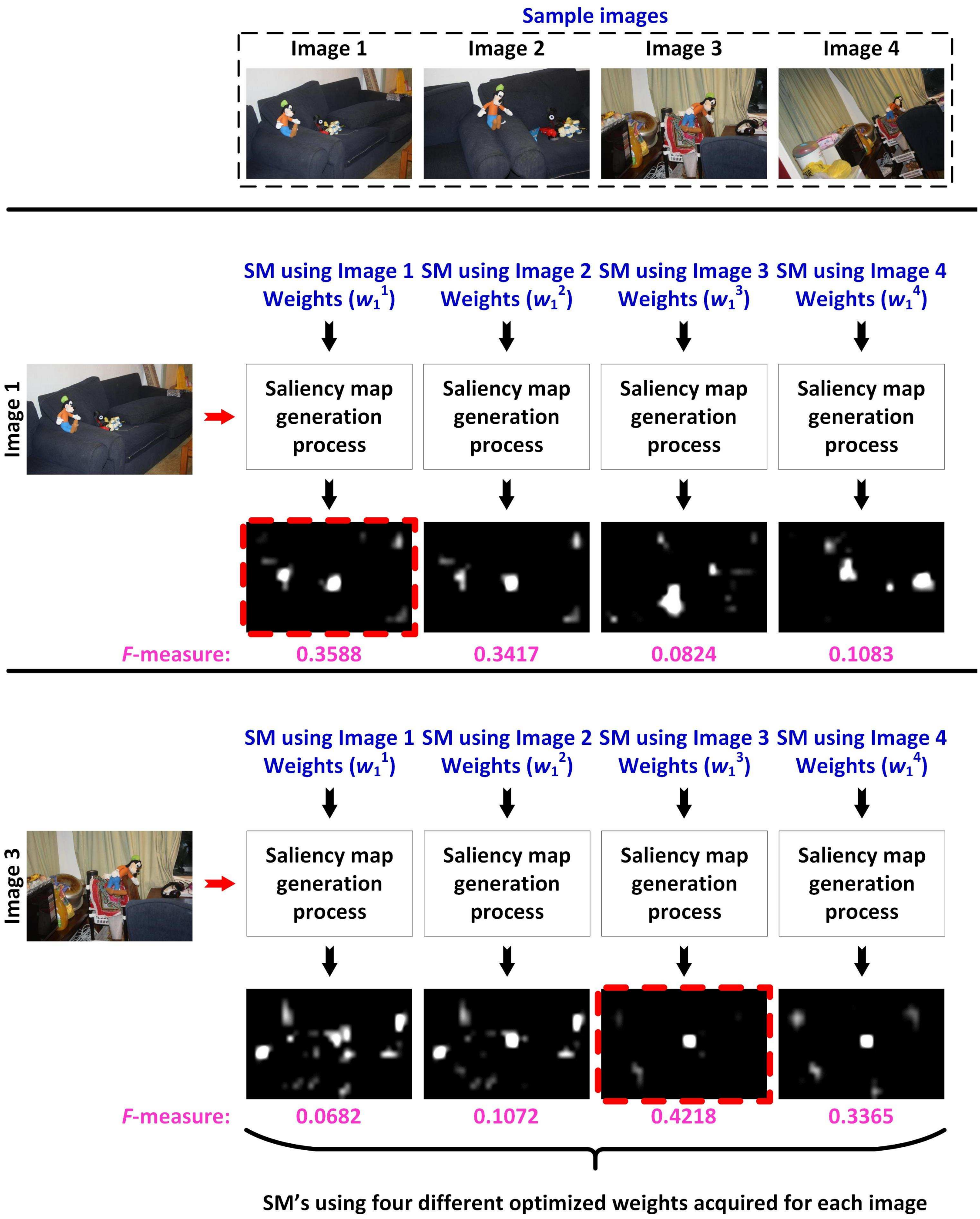

Sample images
Image 1
Image 2
Image 3
Image 4
SM using Image 1 Weights (w_1^1)
SM using Image 2 Weights (w_1^2)
SM using Image 3 Weights (w_1^3)
SM using Image 4 Weights (w_1^4)
Image 1
Saliency map generation process
F-measure:
0.3588
0.3417
0.0824
0.1083
SM using Image 1 Weights (w_1^1)
SM using Image 2 Weights (w_1^2)
SM using Image 3 Weights (w_1^3)
SM using Image 4 Weights (w_1^4)
Image 3
Saliency map generation process
F-measure:
0.0682
0.1072
0.4218
0.3365
SM's using four different optimized weights acquired for each image

Figure 4.4: The impact of using optimized weights learned from contextually different and similar types of images. A sample of four images are used to demonstrate this concept. When an optimized weight vector that is learned specifically for the candidate image (in this example images 1 and 3), the best performance in *F*-measure was achieved (see the highlighted saliency map by a dashed red box). When other optimized weights are applied on these two images, the weights learned for those images that are contextually similar achieved comparable performance to that achieved by the default ones.

ing TDCoW. However, one of the major limitations of that model is the implementation practicality due to the extensive and computationally demanding contextual matching process. As a result, a more efficient way of assigning weights to the image features based on the contextual information is needed.

In this chapter, two types of contextual descriptors are explored. The first descriptor is the gist descriptor proposed by Oliva and Torralba [2] referred to as *envelope gist*. This descriptor provides a structure of the real world scenes through a spatial envelope that is characterized by various scene attributes such as naturalness, roughness, and openness. Modeling of these attributes is carried out using the discrete Fourier transform (DFT) and the windowed Fourier transform (WFT). The descriptor itself is constructed using Gabor filters. Given an input image, a descriptor is constructed by convolving the image with 32 Gabor filters (if we consider four scales and eight orientations) producing 32 feature maps. By default, each map is divided into 4×4 grid and then the average over all feature map values within each grid is performed. If more detailed information is required, the grid size can be decreased for finer contextual extraction at the expense of computational complexity.

Concatenating the 16 averaged values over the grids for all 32 feature maps results in the final descriptor of length 512. Intuitively, this descriptor summarizes the gradient information for different parts of an image and provides a rough description of the scene. An important property of this descriptor is that the envelope attributes ignore local object information contained in a scene. Figure 4.5 shows a sample image and the visualization of its contextual descriptor using the local energy spectrum at each spatial location. Note the similarity in energy spectrum of regions of the image is contextually similar.

The second type of context descriptor is less complex and easier to compute than the *envelope gist* descriptor. Proposed by Rasolzadeh et al. [23], this descriptor basically extracts the low-level attentional features

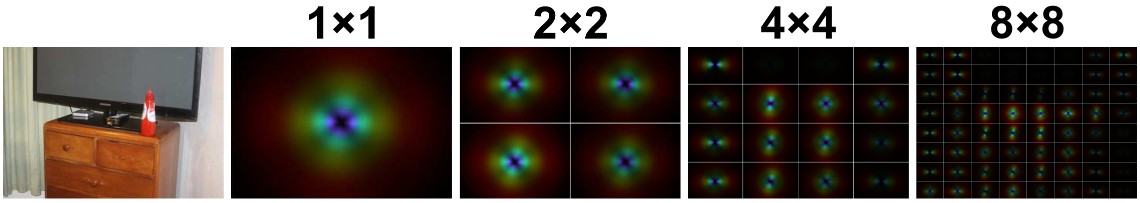


Figure 4.5: A visualization of the Gist descriptor proposed by Oliva and Torralba [2]. As the number of blocks is increased, a more detailed contextual information can be extracted at the expense of increase in the length of the descriptor. The energy spectrogram is constructed from four image scales with eight bin representation at each scale.

for the whole image by providing a global distribution of the low-level features across the entire image. Ideally, the descriptor should be a histogram of the features over all the attentional sub-channels and conspicuity maps. Instead a much simpler approach was considered for efficient computation. The descriptor represents the total energy contained in the feature maps. If the dimension of a feature map is $M \times N$, the the energy content of the feature map is computed as follows:

$$c^i = \sum_{j=1}^{M \times N} ((f_i^j)^2) \tag{4.5}$$

where f_i^j represents a feature map value at location j for the sub-channel feature i. With a total of seven sub-channel features described earlier, this becomes the dimension of this descriptor.

Although this descriptor is computationally efficient, it has a limitation. A feature map describes the spatial saliency location of an image. Equivalently, this descriptor computes the total saliency of a feature map. Hence, the contextual description is coming through saliency description rather than the actual content of the image. To overcome this limitation, we modify this descriptor by computing the energy directly from the filtered features (i.e., before applying the center-surround and normalization mechanisms). This will provide a better description of the image using the raw low-level features. We only compute the energy of the features at three scales for efficiency since the filtering is performed on nine different levels. Hence, the descriptor length becomes $7 \times 3 = 21$. We refer to the contextual descriptor proposed by Rasolzadeh et al. as *attentional gist* and to the proposed modified version as *modified attentional gist*.

To see the descriptive quality of the three contextual descriptors, two sample images are selected from a pool of 22 images and shown in Fig. 4.6. All three contextual descriptors are computed and compared with the respective descriptors for all the images in the pool (excluding the two selected sample images). The comparison is performed by calculating the Euclidean distance between the vector of descriptors. The image that corresponds to the minimum distance is selected as a match to the test images, indicating a contextual similarity. When *envelope gist* is used, we can clearly see that the best match images for the two test images exhibit visual and background similarity. The same can be said about the *modified attentional gist* descriptor. Note that the best matches differ in both cases but still exhibit high similarity to the test images. By contrast, the *attentional gist* on both examples produced matches that are visually unrelated to the test images, supporting the previous analysis that the descriptor is based on the saliency contents of an image rather than its gist content.

The above observation is not only confined to the two sample images but can be generalized to any set of images. The descriptive power of these three contextual descriptors will be evident when the target detection accuracy performance is discussed in the result section.

4.2.1.3 Contextual association

In order to generate the saliency map of a novel image, the set of optimized weights is required. However, because the groundtruth map is unavailable, this weight vector needs to be predicted. For images with groundtruth map (training images), the optimized weight vector and the contextual descriptor for that particular image are acquired. So the objective here is to generate a near-optimal weight vector $\mathbf{w}_1$ for a novel image

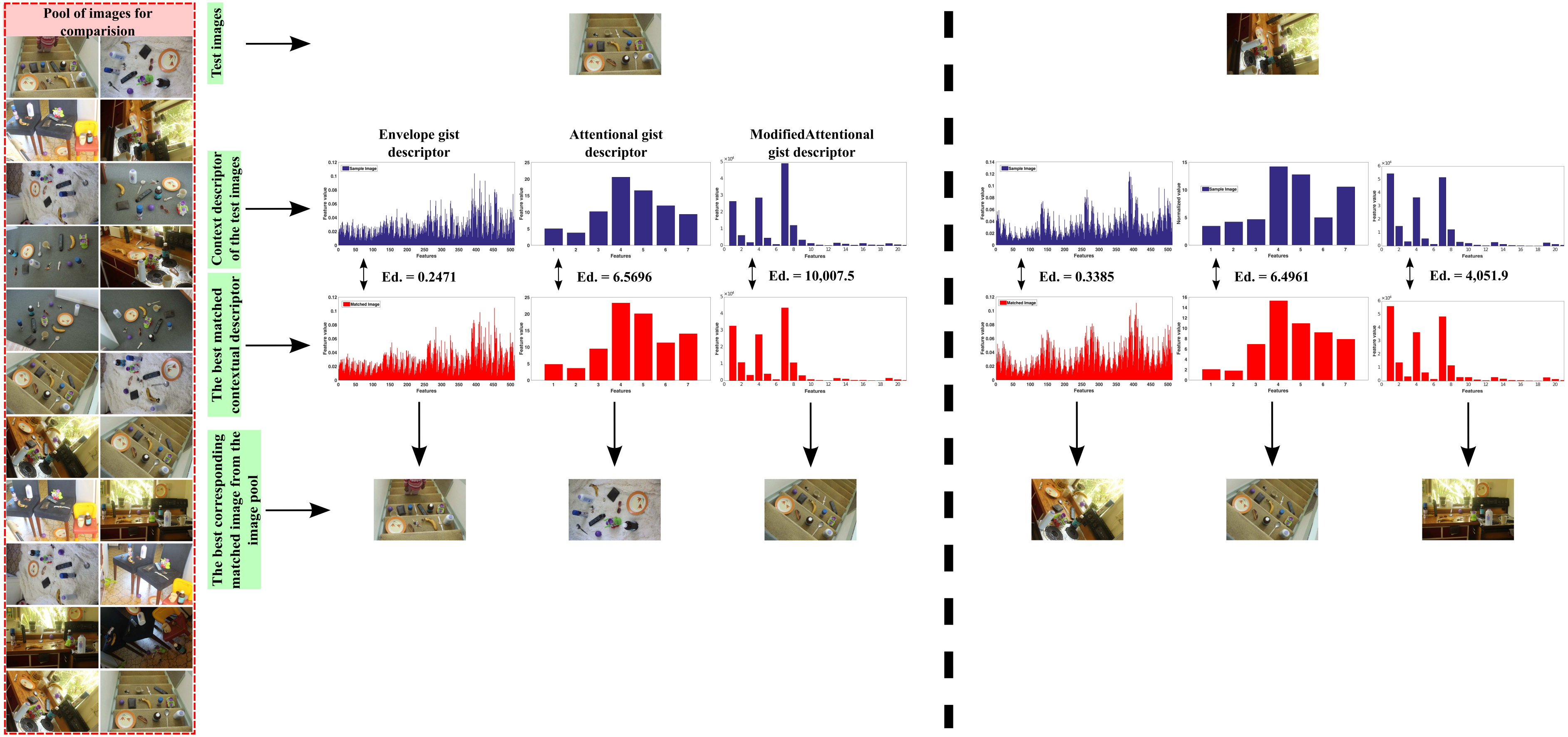


Figure 4.6: Three contextual descriptors' comparison through an example. The two images on the top of the figure are sample images to be contextually matched with a pool of example images at the left of the figure. Both the *envelop gist* and the *modified attentional gist* yielded good match that is visually similar by measuring the Euclidean distance between the descriptors of the test images and the descriptors of all the images in the pool. The *attentional gist* descriptor failed to produce a correct match in both examples despite the corresponding match having the lowest Euclidean distance from the pool.

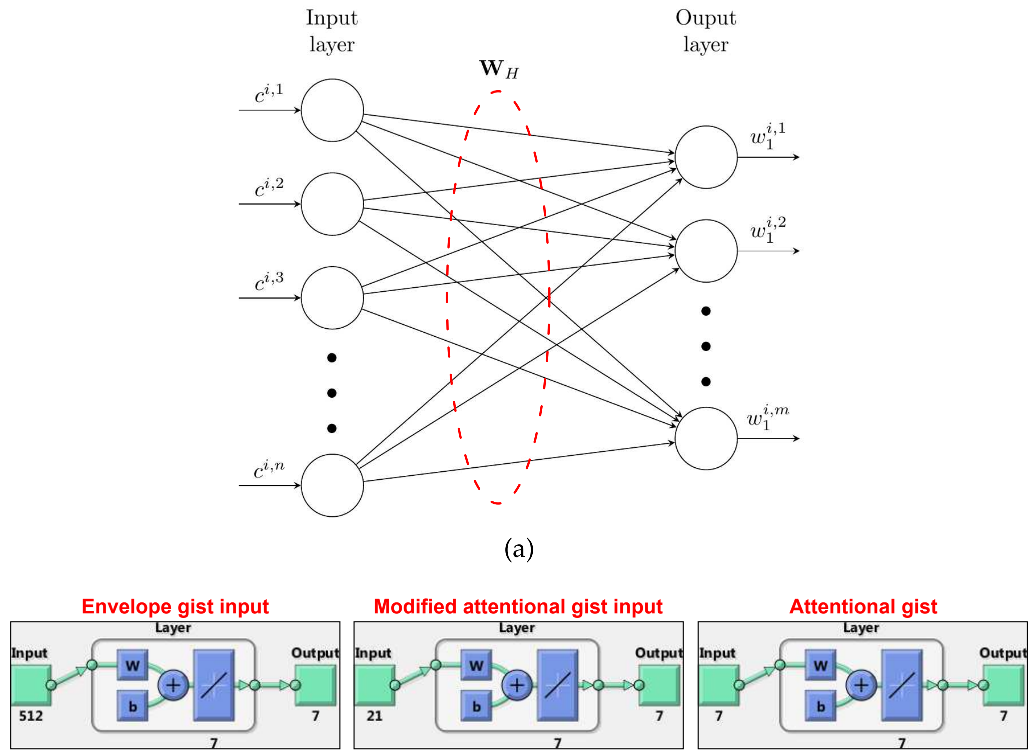
through knowledge of its contextual information.

One way to do so is to compare the context of the new image with the contextual descriptor of the seen images used as training examples. The weight vector associated with the best match is assigned to the new image. This is the approach that was followed when performing contextual feature weighting in TDCoW model through prior clustering in chapter 3. The disadvantage of this approach is that it requires an exhaustive matching with the contextual descriptors of all the training images. The approach becomes computationally demanding when the number of training images is large or when the contextual descriptor has a large dimension. The advantage of using the clustering approach is that it produces accurate matching particularly when the number of clusters is large (best result is achieved when there is one image per cluster). For this reason, in TDCoW model, to have a better matching accuracy was preferred but at the expense of an increment in processing time.

An alternate solution to this problem is to use a pattern learning procedure instead. As it will be evident below, this model associates the input (contextual descriptor) and the output target (weight vectors) in a linear combination manner. This makes the prediction of a weight vector very efficient for a novel image through its contextual descriptor.

For this purpose, a hetero-associative neural network is used to exploit a relation between the contextual descriptor and the optimized weight vector. We use a single layer linear neural network for this purpose where as the inputs to this network represent the contextual descriptor whereas the output represent the optimized weight vector as shown in Fig. 4.7. We refer to this as a hetero-associative network because the two variables are different (i.e., the input and the target to predict represent different entities). The network is trained using Widrow-Hoff weight/bias learning function (also known as the delta or least mean squared (LMS) rule).

The hetero-associative neural network has 512, 21 and seven input units representing the contextual descriptor dimension when *envelope gist*



(a)

(b)

Figure 4.7: The neural network hetero-associative model: (a) A general single layer network with n inputs as contextual descriptor and m outputs as the optimized weights w_1 (b) Three types of inputs to the model corresponding to *envelope gist*, *modified attentional gist* and the *attentional gist*.

with default parameters, *modified attentional gist* and *attentional gist* are used. The number of output units is fixed to seven as shown in Fig. 4.7(b).

The final output of this module represents an optimized weight vector that is tuned contextually to maximize the detection accuracy. When these weights are applied to the feature and conspicuity maps through the saliency generation process, a top-down saliency map is produced which is referred to as *contextual top-down saliency map* (SM_{TDC}). Note that this module does not consider any target based information or features. The next section describes how to utilize target object information for generating another type of top-down saliency map.

4.2.2 Module 2: attention target features (ATF)

In this section, we consider the second source of information (i.e., target knowledge) for learning the weights of the features. This kind of weight learning is different from the one learned in the contextual module as the latter does not consider any target object features while performing the optimization. This weighting is explicitly learned by investigating those features that would best describe the target object. This is achieved by exploiting the variation within feature sub-channels instead of performing an optimization over an objective function.

Figure 4.8 shows the detailed process of generating the target feature based attention weights in a training phase. Note that the dimension of the weight vector is higher than $\mathbf{w}_1$ as the weights are computed over various scales of the feature channels. Inspired by the idea proposed in [143] for acquiring candidate target regions, a BU saliency map is generated for an image through an *Itti* model. When the saliency map hits a target object (i.e., part of the target object with reference to the groundtruth map is detected through the saliency map), L number of locations are randomly sampled from the salient region at the location of the target object region. Since each pixel location of a saliency map is produced through the combination and normalization of various feature maps at different scales, each point can be summarized as a vector of feature values, one from each feature map scale. For the seven feature maps discussed earlier, each feature map is generated through six scales (the default number of scales in *Itti* model). Hence, a single point on a saliency map can be represented by a vector of dimension $6 \times 7 = 42$ of normalized feature values.

For the points that lie within the groundtruth region, referred to as a *view* ($\mathbf{v}$), the accumulation of these points produces the target object view of a single instance image (*instance view* ($\mathbf{V}$)). Similarly by accumulating all such views from K number of images/instances, we establish a target object feature representation that we refer to as *object view* ($\mathbf{O}$) and can be

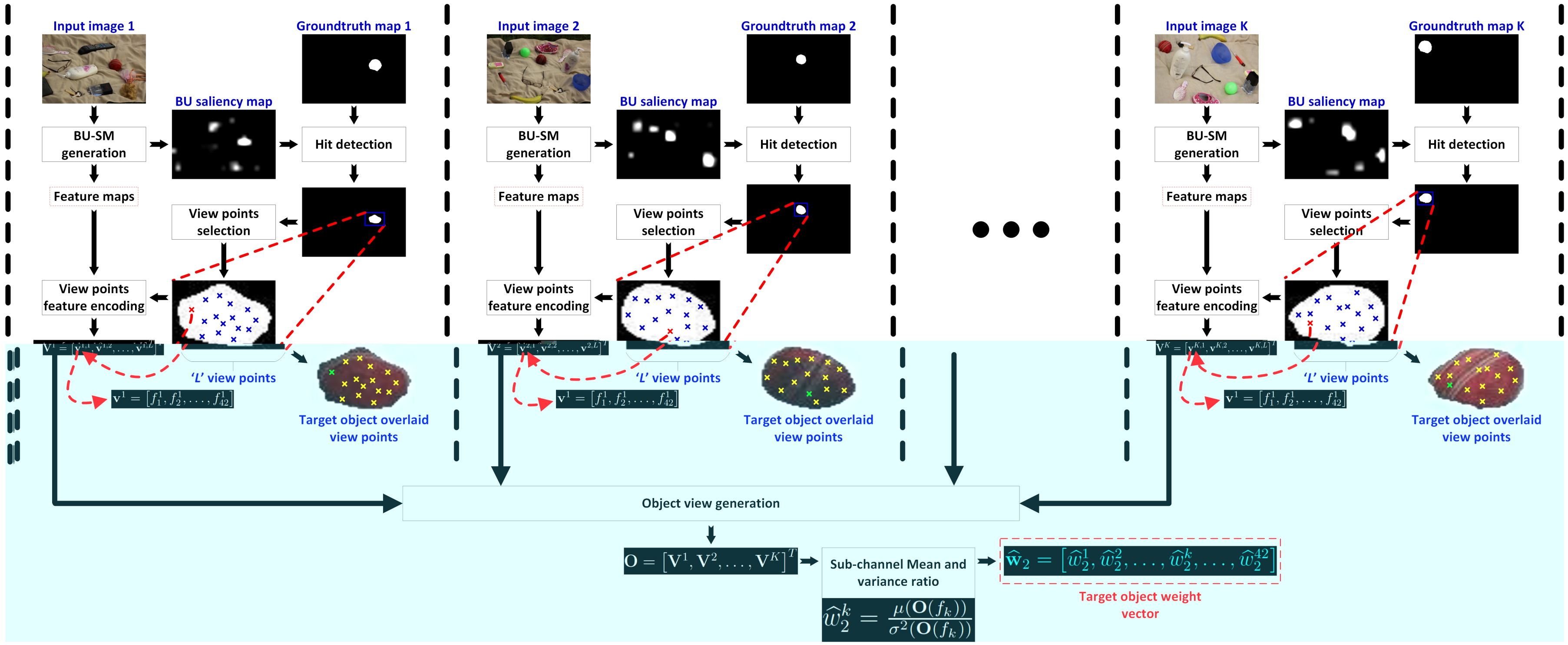

Input image 1
Groundtruth map 1
BU saliency map
BU-SM generation
Hit detection
Feature maps
View points selection
View points feature encoding
$V^1 = [v^{1,1}, v^{1,2}, \ldots, v^{1,L}]$
'L' view points
$v^1 = [f_1^1, f_2^1, \ldots, f_{42}^1]$
Target object overlaid view points
Input image 2
Groundtruth map 2
BU saliency map
BU-SM generation
Hit detection
Feature maps
View points selection
View points feature encoding
$V^2 = [v^{2,1}, v^{2,2}, \ldots, v^{2,L}]^T$
'L' view points
$v^1 = [f_1^1, f_2^1, \ldots, f_{42}^1]$
Target object overlaid view points
Input image K
Groundtruth map K
BU saliency map
BU-SM generation
Hit detection
Feature maps
View points selection
View points feature encoding
$V^K = [v^{K,1}, v^{K,2}, \ldots, v^{K,L}]^T$
'L' view points
$v^1 = [f_1^1, f_2^1, \ldots, f_{42}^1]$
Target object overlaid view points
Object view generation
$\mathbf{O} = [\mathbf{V}^1, \mathbf{V}^2, \ldots, \mathbf{V}^K]^T$
Sub-channel Mean and variance ratio
$\widehat{w}_2^k = \frac{\mu(\mathbf{O}(f_k))}{\sigma^2(\mathbf{O}(f_k))}$
$\widehat{\mathbf{w}}_2 = [\widehat{w}_2^1, \widehat{w}_2^2, \ldots, \widehat{w}_2^k, \ldots, \widehat{w}_2^{42}]$
Target object weight vector

Figure 4.8: The steps involved in generating the target feature weights. The hit detection occurs when the initial BU saliency map of the image coincides with the groundtruth map of that image. The hit decision is decided by applying a spatial threshold value on the overlapped region between the two maps. This threshold value is kept small so that the chances of a success hit is high. Once the hit region is detected, L number of random points are selected as views for the image which is followed by feature vector representation of these views. For various vector notations in the figure, refer to equations 4.6 and 4.7 and their description.

summarized as follows:

$$\begin{aligned}\mathbf{v}^i &= \left[f_1^i, f_2^i, \ldots, f_{42}^i\right] \quad \text{where} \quad i = 1, 2, \ldots, L \\ \mathbf{V}^j &= \left[\mathbf{v}^{j,1}, \mathbf{v}^{j,2}, \ldots, \mathbf{v}^{j,L}\right]^T \quad \text{where} \quad j = 1, 2, \ldots, K \\ \mathbf{O} &= \left[\mathbf{V}^1, \mathbf{V}^2, \ldots, \mathbf{V}^K\right]^T\end{aligned} \tag{4.6}$$

where f is a sub-channel feature value. Note that the features' indices in Eq. (4.6) are in sequence with respect to the scale number. For instance, $f_1, f_2, \ldots, f_6$ correspond to the feature map values of *R/G* computed at scale one to six respectively. Similarly $f_7, f_8, \ldots, f_{12}$ is for the *B/Y* feature map at the same six scales in order and so on for the remainder of the features.

Once the *object view* **O** is generated for a particular target object, the mean μ_k and variance σ_k^2 of its sub-channel feature maps are computed independently where k is the index of the sub-channel at some scale as explained previously. A sub-channel feature is hypothesized to be relevant if its mean feature is high while the variance is low. For a target object, a high mean of a sub-channel is desirable to generate a saliency map that has high activation at the target object location. If the variance is high, then this is an indication that the sub-channel feature is inconsistent in producing good activation points within the target region. High variance suggests that the sub-channel feature is not a good representative of the target object and less relevant. Concretely, a sub-channel feature weight is computed by taking the ratio of the two factors as follows:

$$\begin{aligned}\mathbf{w}_2 &= \left[w_2^1, w_2^2, \ldots, w_2^k, \ldots, w_2^{42}\right] \\ w_2^k &= \frac{\mu(\mathbf{O}(f_k))}{\sigma^2(\mathbf{O}(f_k))}\end{aligned} \tag{4.7}$$

where $\mathbf{w}_2$ represents the target object weight vector. Note that the length of this weight is greater than that of $\mathbf{w}_1$ as describing the objects requires more fine grained in-depth features and this is achieved by gathering feature information at various scales. In addition, $\mathbf{w}_2$ represents a generic single weight vector tuned for a specific target object and can be applied to any image for detecting the same target object.

To assign conspicuity map weights, the maximum weight value over all sub-channels belonging to the same conspicuity feature channel is selected. This criterion guarantees that a conspicuity feature receives importance if any of its sub-channel features at any scale is important and vice versa.

4.2.3 Map combination

The saliency maps generated by the two modules are combined before being processed by the RTF module. Two combination approaches are followed which are given as follows:

$$SM_{TD}^{C} = \frac{SM_{TD}^{w_1} + SM_{TD}^{w_2}}{2} \quad \text{or} \quad SM_{TD}^{C} = AND(\overline{SM_{TD}^{w_1}}, \overline{SM_{TD}^{w_2}}) \tag{4.8}$$

where$\overline{SM_{TD}^{w_1}}$ is the saliency map generated by the first module after converting it into a binary map using a threshold value and $\overline{SM_{TD}^{w_2}}$ is for the second module. The way the two saliency maps are combined will not only have an impact on the quality of the generated combined saliency map SM_{TD}^{C} but will also affect the final output of the RTF module. The first combination strategy assumes that both saliency maps highlight saliency regions that differ spatially. Hence, to incorporate all these regions into final saliency map, the arithmetic mean is used. The benefit in doing so would be to have a better precision over the detected target object region. On the other hand, the second combination strategy which is based on performing logical *AND* operation assumes that the common saliency region in both maps are more likely to contain the target object whereas the remainder of the saliency regions represent false positives.

4.2.4 Module 3: recognition target features (RTF)

This module acts as a post-processing step of recognizing the target object and can be considered an optional step for improving the detection accuracy of the target object. The input to this module is the combined map generated by the other two modules. The combined map provides candidate regions of interest that are likely to contain the target object. The purpose of this module is to reduce the number of false positive regions established by the combined saliency map through classification. For a given saliency map, the module pays attention to each candidate region, extracts some features and classifies it as belonging to the target object class or a non-target class.

The sequence of visiting the attention region becomes unimportant if the purpose is to exhaustively process all the regions. For an active vision application, it is preferable to attend to regions that are highly salient or to apply a winner-take-all and inhibition-of-return for progressive and dynamic attention selection. Since the objective here is to eliminate false positive regions for a better saliency map, the sequential approach is applied. Note that if the combined saliency map fails to sample the target region, then the recognition module becomes ineffective. Hence this module is dependent on the target object detection accuracy by the preceding process of weight selection.

Referring to Fig. 4.1, the module directly learns target object representation through features that are extracted in the ATF module. As it is obvious in Eq. (4.6), a target object can be represented as a combination of the views of its various instances which is summarized by **O**. As depicted in the ATF module, **O** is created from K images that contain the target object during the learning phase. All such views and the combined **O** matrix represent a positive class $c_j = 1$ feature matrix. Furthermore, from the same K images, a negative class feature matrix is constructed by gathering random views of non-target regions. To make an equal number of views from both classes, $L \times K$ number of views construct the negative class feature matrix represented as $\mathbf{N}$ where the views of this negative class class $c_j = 0$ follow the same feature pattern as $\mathbf{v}^i$ given in Eq. (4.6). Note that the views of the negative class are diverse as they cover different scene contents including background, non-target objects, and distracting objects. This could result in an increase in the misclassification rate of the negative views. However, this is a generic approach that can be improved by fine-tuning the negative class into sub-classes. All the experiments related to this module are performed over two classes (i.e., $j = 1$ and 2) only.

To learn a classifier, various assumptions are made to simplify the problem. First of all, it is assumed that each view from either class is equally likely. Secondly, for simplification, the sub-channel features at the scales mentioned earlier are assumed to be independent. Although this could be true for the sub-channels features (e.g., orientation at 45° is independent from *R/G* contrast), the independence of scale features belonging the same sub-channel might be arguable. If the dependency factor is considered, more computation of joint probabilities and other dependency parameters need to be considered. Hence to avoid that, we assume interdependency at all scales and sub-channel features. Finally, for a given class and as an approximation, it is reasonable to consider the sub-channel features as random variables that follow a normal distribution. If X_k is a random variable of sub-channel features (f_k belonging to either $\mathbf{O}$ or $\mathbf{N}$ and approximated by a normal distribution $X_k \sim \mathcal{N}(\mu_k,\, \sigma_k^2)$ where μ_k and σ_k^2 represent the sub-channel feature mean and variance respectively, then for X_k to belong to a class $C = c_j$ where as before c_j is either zero or one for binary classification, we use the Bayes rule as follows:

$$\Pr\Big(C|X = (X_1, X_2, \ldots, X_{42})\Big) = \frac{\Pr\Big(X = (X_1, X_2, \ldots, X_{42})|C\Big)\Pr\Big(C\Big)}{\Pr\Big(X\Big)} \tag{4.9}$$

Ideally the likelihood function $\Pr(X|C)$ should follow a multivariate Gaussian distribution with a sub-channel feature mean and covariance matrix. However, with the feature interdependency assumption, each subchannel feature is treated as a separate univariate Gaussian distribution and given as:

$$\Pr\Big(X_k|C\Big) = \frac{1}{\sigma_k\sqrt{2\pi}}e^{-(x_k-\mu_k)^2/2\sigma_k^2} \tag{4.10}$$

Note that the independence assumptions made above may affect the classification accuracy as it would fail to exploit certain structures in the feature space but it would not be that significant.

To predict the posterior probability $\Pr(C|X)$, Naive Bayes classifier is used as it perfectly matches the interdependency assumption imposed on the sub-channel features. So equation 4.9 can be written as:

$$\Pr\Big(C|X\Big) = \frac{\Pr\Big(C\Big)\prod_{k=1}^{42}\Pr\Big(X_k|C\Big)}{\Pr\Big(X\Big)} \tag{4.11}$$

Hence the objective of the Naive Bayes classifier is to maximize the posterior probability over the two classes:

$$\arg\max_{C}\Pr\Big(C|X\Big) = \arg\max_{C}\Pr\Big(C\Big)\prod_{k=1}^{42}\Pr\Big(X_k|C\Big) \tag{4.12}$$

Upon generating the saliency map of a novel image, each obtained saliency region is processed for the likelihood of that region to contain the target object. Various views are extracted from each region of the saliency map to be classified by the learned Naive Bayes model. Unlike in the training phase where L views were extracted, here all the views/points are extracted and classified instead. Note that each view is classified independently by the classifier. Although the model learns an object level feature distribution, it classifies on the basis of individual views. Because there is a possibility that a candidate region sampled by the saliency generation process contains view from both negative and positive classes, combining both views for an object level classification would create confusion to the classifier as it would be trained only on purely negative and purely positive object views.

Once all the view points at a saliency region are classified using Eq. (4.12), a decision is made on the attended region based on the predicted labels of its views. The first decision approach shown in Fig. 4.9 is based on the majority ranking of the views. If $\mathbf{B}$ represents a vector of predicted class labels of all the views within an attended region in a saliency map, then the majority ranking decision can be written as follows:

$$\widehat{B} = \begin{cases} c_1, & \text{if } |\mathbf{B}^{c_1}| \geq |\mathbf{B}^{c_2}| \\ c_2, & \text{otherwise} \end{cases} \tag{4.13}$$

This approach is useful in eliminating false positive regions that were selected by the saliency combination process. However, one disadvantage of this approach is that it does not discriminate between high and low view separation ratios. For instance, if $|\mathbf{B}^{c_1}| = |\mathbf{B}^{c_2}| - 1$, then the majority ranking will label the region as negative class and would eliminate that region. It is possible that this highlighted region contains both the target and non-target views. Because the number of views belonging to non-targets is slightly greater than that for the target, the region is eliminated and hence the precision is degraded.

To handle the above mentioned problem, another decision approach is used that is based on weighting the region by the number of views being classified as positive as follows:

$$w_R^u = \frac{\textit{Number of positive views}}{|R^u|} \quad \text{where} \quad u = 1, 2, \ldots, S \tag{4.14}$$

where S is the number of salient regions in a saliency map and $|R^u|$ is the total number of views in the u^{th} saliency region. The obtained weights are associated to each region of the saliency map to yield the final top-down saliency map as demonstrated through an example shown in Fig. 4.10. This approach does not eliminate the region if the number of predicted view label is low. However, it penalizes it by weighting the region with a small weight. A limitation of this approach is that if a region is a non-target and has been weighted by a low weight, it will still be considered as

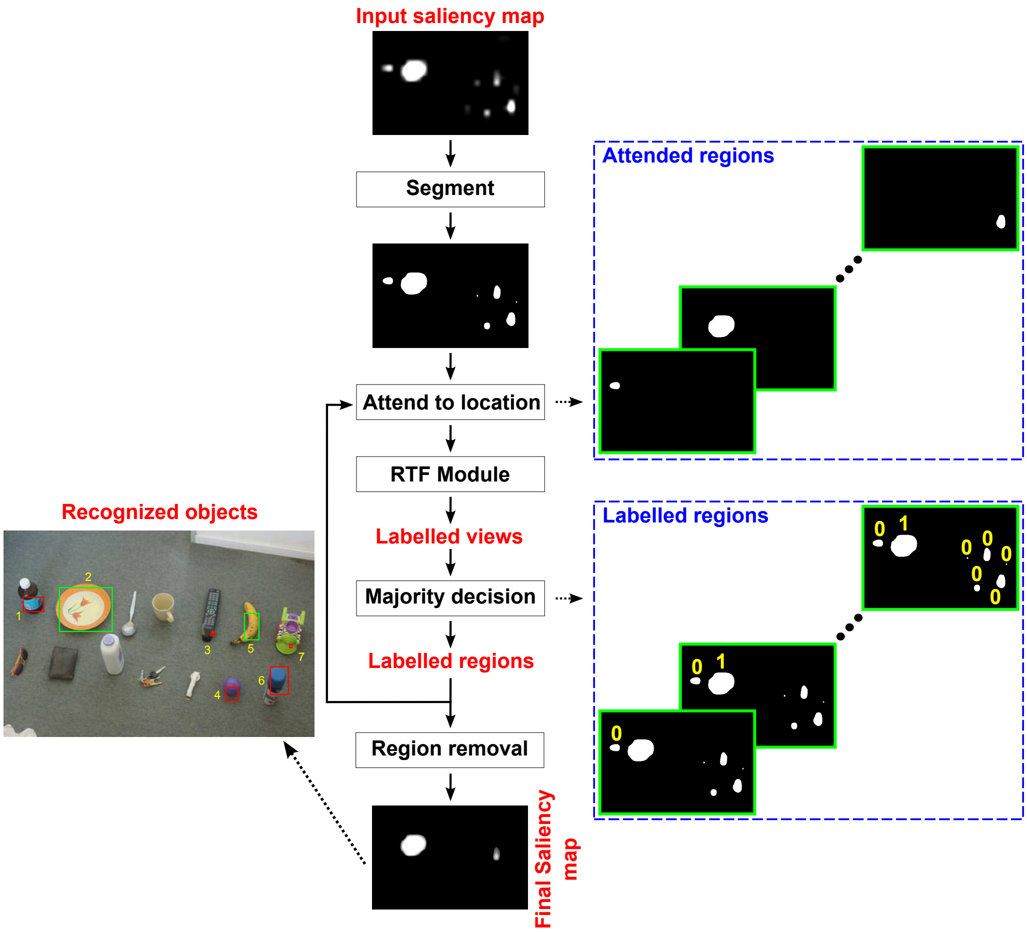


Figure 4.9: The majority ranking decision making for generating the final top-down saliency map. Each attended location is recognized by the RTF module by classifying the views within that region to either target or non-target class. The majority ranking decision labels an attended region based on the majority class of the views within that region. Non-target classified regions are eliminated from the final saliency map.

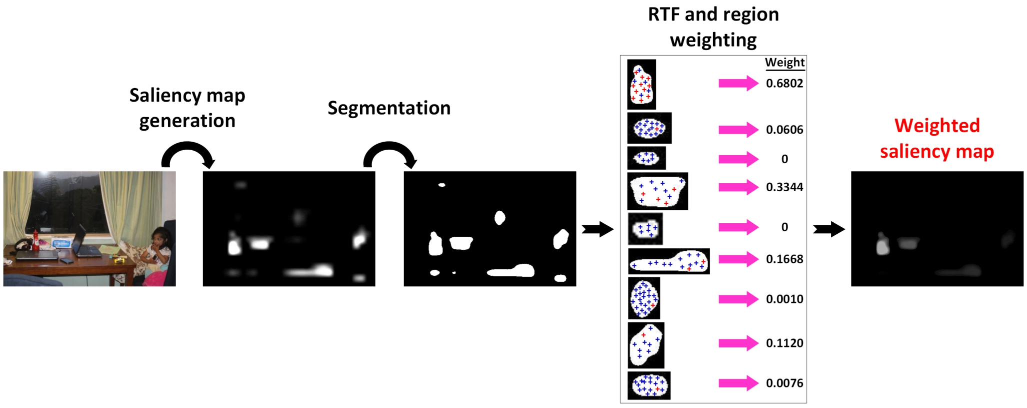


Figure 4.10: The region weighting decision making for generating the final top-down saliency map. As in majority ranking, the RTF module performs view classification of each attended location. The weighting decision assigns weights to each region which is computed as the ratio between the number of positive views being classified and the total number of views within region. The number of views in each region varies depending on the size of the region. Note that in the actual RTF model, exhaustive view extraction takes place from each region whereas in the figure, and for visualization purpose, sample views are considered. The red $'+'$ sign indicates a positive view.

a region containing the target but with small probability. Hence, there is a trade-off between the two approaches and so both are used to evaluate the performance of our proposed model.

4.2.5 Training and testing phases

This section summarizes the functionality of the three modules from the perspective of training and testing phases. From a dataset of $2 \times K$ images, K images are selected for training various parameters of the modules as follows:

1. For each image from the training set, learn an optimized set of attention feature weight vector $\mathbf{w}_1^i$ where $i = 1, 2, \ldots, L$ and the length of $\mathbf{w}_1^i$ is $D = 10$. Seven of these are the weights for the sub-channel FMs and three for the CMs. The weight learning is performed using PSO with the objective/fitness function is to maximize the *F*-

measure score between the SM of the image that is generated by this weight vector and its groundtruth map.

2. Compute the contextual information of each image using one of the three contextual extraction techniques provided in section 4.2.1.2 to produce a contextual descriptor $\mathbf{c}^i$ of length U.

3. Form K pairs of examples consisting of the optimized weight and the contextual descriptor (i.e., $(\mathbf{c}^i, \mathbf{w}_1^i)$ and feed it to a hetero-associative single layer neural network that learns an association between the input and the output variables.

4. For each of the K training images, compute the sub-channel features at six different scales to generate the BU saliency map. Each activation point on the saliency maps is a vector of length 42.

5. Compare the saliency map with the groundtruth map. If a hit is detected, extract L sample activation points (called views) to construct a target feature descriptor of dimension $L \times 42$. Combine this data with similar views from other examples to construct an object view as given in Eq. (4.6).

6. Compute the sub-channel feature mean and variance of the object view and take their ratio using Eq. (4.7) to construct a target feature weight vector $\mathbf{w}_2$ of length 42.

7. Perform the previous step but now on various negative (non-target) views of the training images.

8. Represent the two set of views (i.e., belonging to either negative or positive class) through a univariate Gaussian distribution for each sub-channel feature using Eq. (4.10).

9. Assuming independence of the feature channels, learn a Naive Bayes binary classifier from the distributions obtained from the previous

step using Eq. (4.11) to maximize the classification accuracy using the objective function given in Eq. (4.12).

For a novel image from the testing dataset, the final top-down saliency map denoted as SM_{TD} that highlights regions of the image that are most likely to contain the target object is generated as follows:

1. Compute the contextual descriptor of the novel image using one of the three techniques mentioned earlier.

2. From the learned hetero-associative model (should be learned on the same contextual descriptor used in the testing phase), compute the optimized set of weights $\mathbf{w}_1^{opt,new}$.

3. Compute the saliency map through *Itti* model through weighting the feature and conspicuity maps with the learned $\mathbf{w}_1^{opt,new}$ weight vector from the previous step. The generated saliency map from this step is denoted as $SM_{TD}^{w_1}$.

4. Compute the target object based saliency map denoted as $SM_{TD}^{w_2}$ by again weighting the attentional features of the *Itti* model but now with the learned target object feature weight vector $\mathbf{w}_2$.

5. Combine both saliency maps obtained from steps three and four to form the combined saliency map denoted as SM_{TD}^{C} using one of the combination operations given in Eq. (4.8).

6. Segment SM_{TD}^{C} using mean-shift-segmentation approach to generate a binary image containing candidate regions that likely to contain the target.

7. Attend to each of these regions separately in any sequence. Once a region is attended to, views are extracted in the same way as in the training phase but now for every activation point within the attended region.

8. Classify the views individually to either belonging to the target or non-target object using the learned Naive Bayes classifier.

9. Apply either a weighting (using Eq. (4.14)) or majority ranking (using Eq. (4.13)) approach to the classified views from each attended location to reduce or eliminate the false positive regions and ultimately generate the final top-down saliency map SM_{TD}.

4.3 Experiment design

4.3.1 Dataset

To validate the performance of the proposed model, the model is evaluated on seven self-created datasets that vary in terms of the target object to be detected, background, object view, illumination and clutterness. Since the proposed model is an attention based, these datasets have the characteristic of variation in the degree of saliency of the target object which lacks in many object recognition, detection, and visual search datasets [144]. Each dataset contains 100 indoor images with varying background complexity. Table 4.1 gives a brief description of each dataset.

The last dataset comprises of 14 objects arranged randomly with either uniform or cluttered background. In some instances, the objects are partially occluded by other distracting objects. Six objects are selected as target objects to be searched for. These objects are listed in Table 4.1 and represent objects that have certain characteristics. For instance, *DS*-7(Spoon) represents an object that is distinguished by its colour as its main characteristic feature. In *DS*-7(Plate) the object size is large compared to other objects and exhibit typical orientation characteristics. For a part-based object, *DS*-7(Train) has an object with multiple parts with different colour and visual characteristics. The diversity in *DS*-7 objects establishes a framework to study the effectiveness of the ATF and RTF modules in describing the target object through primitive attentional features. The groundtruth

Dataset	Target object	Background type	Description
DS-1	Red cricket ball	Simple white uniform	• Illumination conditions does not vary. • The saliency of the target is high compared to that of other objects in the scene. • Contains non-overlapping distracting objects that are positioned at a random distance from the target object. Some of the distracting objects have visual characteristics similar to that of the target object. • Size of the target remains constant in most of the images but the view angle varies
DS-2	Pink soft ball	Cluttered and complex	• Illumination conditions vary considerably. • The saliency of the target is highly variable to that of other objects in the scene. • Contains many distracting objects but do not overlap with the target object. The distracting objects vary from image to another and positioned randomly. Some of the distracting objects have visual characteristics similar to that of the target object. • Size and view of the target object vary considerably. • More than one object instance is present for this target object in some of the images.
DS-3	A piece of corn	Cluttered and complex	• Illumination conditions vary slightly. • The object is always placed inside the fridge compartment, however, the background is highly cluttered and dense. • The saliency of the target is highly variable to that of other objects in the scene. • Contains many distracting objects but in close proximity to the target object. • Size and view of the target object varies considerably. • Occasionally the object is occluded.
DS-4	A black spoon	Cluttered and complex	• Illumination conditions vary considerably. • The object is small and has a slim structure that results in being lost within the cluttered background. • The target object has a very low saliency. • Size and view of the target object vary considerably. • Occasionally the object is occluded.
DS-5	A red ketchup bottle	Cluttered and complex	• Illumination conditions vary considerably. • The target object has a moderate to high saliency. • Size and view of the target object vary considerably. • The object is not occluded but various distracting objects sharing the same colour are present in the images.
DS-6	Goofy Disney character toy	Cluttered and complex	• Illumination conditions vary considerably. • The target object has multiple parts with different colour and geometric characteristics. • The saliency of the object as well as its parts is highly variable. • Size and view of the target object varies considerably. • The object is partially occluded in many images.
DS-7	Multiple objects	Both cluttered and uniform	• There are six objects: banana, plate, remote control, powder bottle, wallet, toy train • The dataset is same for all the objects. • The illumination conditions vary considerably. • All six objects are present in an image but the search is performed for a single target object. Apart from that, 8 more objects are always present in an image acting as distracting objects. • The placement of all 14 object (i.e., target and distractors) is done either systematically or randomly. • The visual complexity and hence the detection of the objects vary. • The saliency of the objects is highly variable. • In some instances, the background is simple (uniform) and in other instance, the background is highly cluttered and complex. • Size and view of the target object varies considerably. • Most of the target object are not occluded.

Table 4.1: Description of the datasets used in this chapter.

maps for these datasets represent fully segmented binary maps that highlight the target objects.

4.3.2 Parameter values

All experiments are performed under the parameter values summarized in Table 4.2. Each experiment is carried out 50 times (splits) and the mean performance is reported along with the standard error in the sample mean. In each split, a random selection of training/testing images from a dataset is considered. Repeating the experiments over multiple splits provides a confidence about the validity of the obtained results.

4.4 Result and analysis

In this section, target detection accuracy and the computation efficiency of the proposed model is presented.

4.4.1 Contextual weighting performance

In this section, we analyze the contextual module of the model from two perspectives. The first is the optimization accuracy of PSO in producing an optimum or near optimum set of weights. Secondly, we perform a detection accuracy comparison of the three contextual descriptors discussed previously. The performance of this module depends on three factors, the optimized weight vector learned through PSO, the contextual descriptor, and the neural network associative model.

For the first factor, to see how optimum the learned weight vectors are, the average F-measure score of the saliency maps generated by these optimized weight vectors is compared to that generated by random weight vectors (we refer to them as brute-force weight vectors). The F-measure score, which is also used as the fitness value of PSO, is indicative of the

Table 4.2: The simulation parameters used throughout the experiments in this chapter.

Simulation Parameter	Value
PSO population size	1000
PSO maximum number of iterations	100
PSO inertia weight	0.768
PSO acceleration constants	$c_1 = c_2 = 1.49$
PSO number of runs	40
PSO fitness function	$\arg\max_{w_1}$ *F-measure*(SM_{w_1})
Number of experiments (splits)	50
Random brute-force combination	$5,000,000$
Number of recognition classes	Binary (2)
Number of datasets	six and one with six objects
Train/test ratio	0.5
F-measure β^2 value	0.3
Candidate region hit ratio (T_1) and (T_2)	0.1 and 0.5 respectively
Percentage of view points per region	70%

detection accuracy of the target object. A common value for β^2 found in the saliency literature when computing *F*-measure is 0.3 [74].

In the random brute-force strategy we initially generate random feature weight vectors such that the weight values are between zero and one. Because it is impossible to perform an exhaustive search over the feature space due to the continuous nature of the weights, a very large number $(5,000,000)$ of these random weight vectors are chosen to approximate a true brute-force strategy.

A saliency map is generated from each weight vector. This is followed by calculating an *F*-measure corresponding to the generated saliency map. Hence, for a single image in a dataset, we have a total of $5,000,000$ *F*-measure scores, one for each random feature weight vector. The one with the highest *F*-measure score is selected. The corresponding weight vector represents the best possible feature weight combination achieved for that image. The process is repeated for the rest of the images in a dataset. The average *F*-measure score provides the target detection accuracy for that dataset when the random-brute-force is applied. This score is compared to that achieved by the optimized weight vector learned through PSO.

Generating large number of random weight vectors to some extent replicate a brute-force combination of weight vectors. Hence, we would expect that the brute-force approach yield good performance. Hence, the objective of this experiment to see how good the PSO learned weights are compared to brute-force approach. In PSO, only 1000×100 weight combinations are used whereas 50 times more weights are used in the brute-force approach. In addition, we have used $100,000$ sample size to establish an equal number of evaluation samples. This is because PSO optimization is carried out by evaluating the fitness value of $100,000$ particles in total (see Table 4.2 for the simulation setup).

As shown in Fig. 4.11, we can see that PSO has achieved a consistently comparable performance in finding the optimized set of weight vectors when compared to the average *F*-measure score achieved by the brute-force strategy with $5,000,000$ samples, despite the lower number of weight vectors being used in PSO than brute-force. The performance gain (i.e., ratio between the *F*-measure of brute-force with $5,000,000$ samples and the optimized approach using PSO) is almost constant across the datasets. In some of the more difficult datasets in which searching for the target object is more demanding (e.g., *DS*-4), the optimized weight vector has a better performance than the random brute-force approach with $5,000,000$ samples. Infact, when equal number of samples were used in the brute-force approach (i.e., $100,000$ samples), PSO outperforms the brute-force across all the datasets. Hence it is concluded that the optimized learned weights through PSO are highly optimized (but perhaps not optimal) for performing the association.

Next, we investigate the descriptive power of the three contextual descriptors and accordingly, select the one that is most suitable for our model. In order to achieve this, the *F*-measure is computed between the saliency map $SM^{w_1}_{TD}$ and the groundtruth map over the test images for all the splits for each descriptor. The obtained *F*-measure profile shown in Fig. 4.11 gives an insight on the detection accuracy of these descriptors.

The *y*-axis in Fig. 4.11 represents the average *F*-measure score over the entire splits. This average score is obtained as follows,

- For each test image in a split, a saliency map is generated using the feature weights generated by one of the five approaches (indicated by a different colour bar in Fig. 4.11).
- The generated saliency map from the above step is compared with the groundtruth map to yield an *F*-measure score.
- The *F*-measure scores of all the images in a split are averaged. This represents a single sample of the population on which a statistical operation is performed.
- The mean value of the average *F*-measure scores obtained from the splits is shown in Fig. 4.11.
- The figure also shows the standard error in the sample mean by setting the confidence interval to 95% using *t*-distribution. By inspecting the distributions of the average *F*-measure scores, we found that they approximate a normal distribution and they are not highly skewed.

The five weighting approaches used in Fig. 4.11 are the brute-force, ideal PSO optimized weight and the learned weights when using *envelope gist*, *modified attentional gist* and *attentional gist*. To analyze the detection accuracy of the three contextual descriptors, we compare their average *F*-measure scores to that obtained when applying ideal weights from PSO.

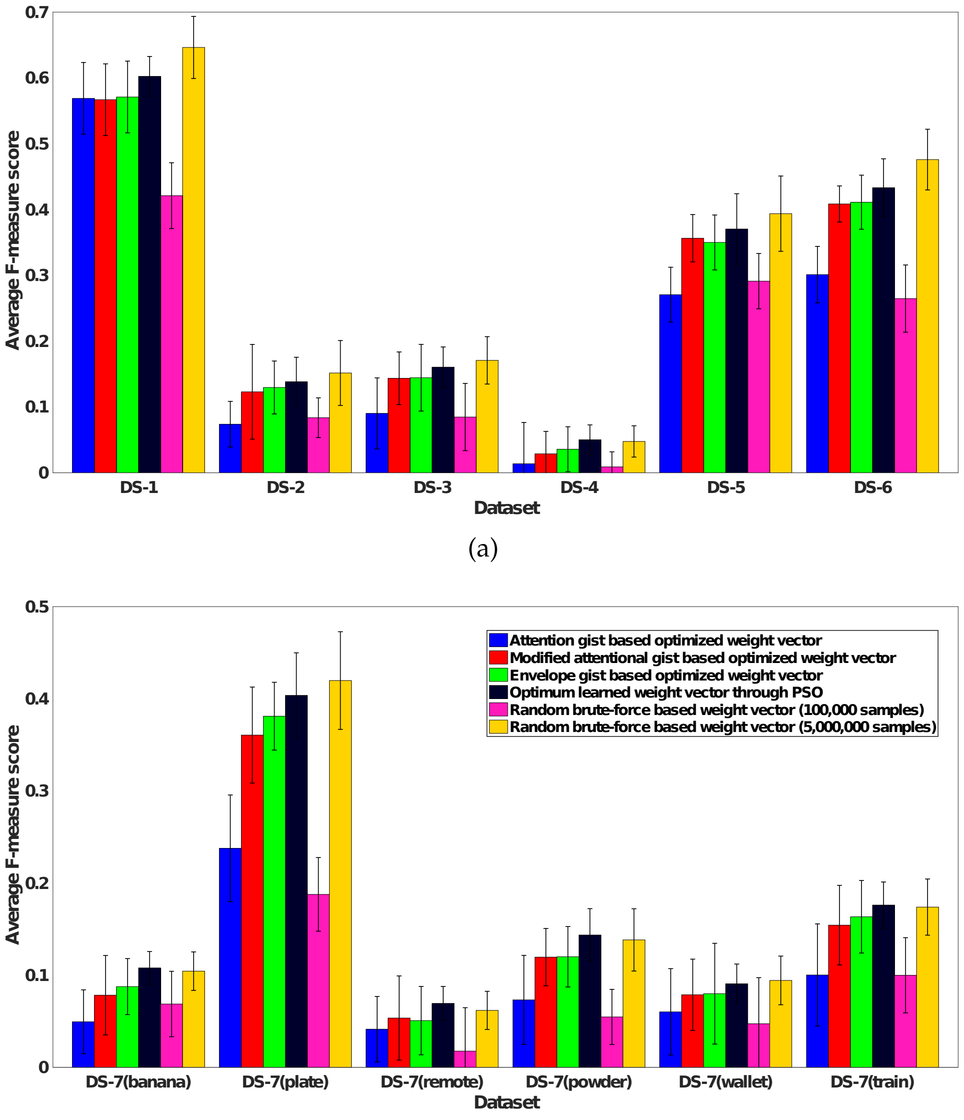
The *envelope gist* descriptor has higher *F*-measure value than both *attentional gist* and the proposed *modified attentional gist* descriptor on all datasets except *DS*-1 (the reference point is the *F*-measure score generated by the ideal PSO based weighting). *attentional gist* has the lowest score compared to the scores of the two other descriptors apart from the *DS*-1 dataset in which it has comparable performance. Finally, the *F*-measure scored by the proposed *modified attentional gist* descriptor is slightly lower than the *envelop gist* descriptor.

Hence, either *envelope gist* or *modified attentional gist* descriptors represent possible contextual choices that would yield a good detection accuracy performance. Although *envelope gist* has slightly better performance than *modified attentional gist*, the latter is more computationally efficient. For this reason, we prioritize efficiency at the expense of slight degradation in detection accuracy performance by choosing *modified attentional gist* in our model for contextual information extraction.

Once the contextual descriptor is extracted from a novel image, the hetero-associative model makes a best effort to predict a weight vector with least mean square error (i.e., between the predicted and the actual groundtruth weight vector). Concretely, if the contextual descriptor of a test image is exactly the same as one of the contextual descriptors used in the training phase, the hetero-associative model will output the exact optimized weight vector. However, if the mapping between the contextual descriptor and the predicted weight performed by the hetero-associative model is such that it takes it far from the actual optimized weight, the prediction error will increase.

Hence, a key element in the hetero-associative model is the contextual descriptor upon which the prediction is made. Since, both *envelope gist* and *modified attentional gist* descriptors describe the images through their holistic contents, they outperform *attentional gist* in detection accuracy as the latter provides less descriptive details. To elaborate this assertion, an experiment is performed that inspects the quality of the produced weights by these descriptors.

In all the datasets, except *DS*-1, while acquiring the images (100 in each



(a)

(b)

Figure 4.11: Target detection performance as characterized measured by the average *F*-measure score of the saliency maps that are generated when different contextual descriptors are used. In addition, the accuracy of the optimized weight vector learned through PSO is compared with the weights obtained through random brute-force weight assignment to show the efficacy of PSO based feature weighting. The results are shown for a) the datasets *DS*-1 to *DS*-6 and b) *DS*-7 dataset with six different objects.

dataset), explicitly 10 types of visually similar content categories were created. In other words, the first 10 images of a dataset exhibit similar content i.e., background but with different rotation, view, illumination and distance from the image acquisition equipment. For the second 10 images, the background becomes different from the first category while establishing the same variations as in the first 10 images and so forth for the other categories. Each image belonging to a certain category is labelled as groundtruth.

With the above contextual level groundtruth information, we perform the following,

- For each test image in a split, its contextual information is extracted using one of the three descriptors.
- This contextual descriptor is fed to the learned hetero-associative model to predict a weight vector.
- The obtained weight vector is compared with the optimum set of weight vectors that were learned through PSO excluding the test image optimum weight. For comparison, we use normalized Pearson correlation.
- The image corresponding to the best match (i.e., highest correlation coefficient) is selected.
- The contextual label of the selected image is identified.
- The above steps are repeated for all the test images for the entire split.

The final output of the above steps is a set of contextual labels for all the selected images. These labels when compared to the groundtruth contextual labels establish a success accuracy S defined as:

$$S = \frac{\text{Number of correct labels}}{\text{Total number of test images}} \tag{4.15}$$

Table 4.3: Success accuracy when using the three descriptors shown for six datasets.

	Dataset					
Descriptor	*DS-1*	*DS-2*	*DS-3*	*DS-4*	*DS-5*	*DS-6*
Envelope gist	100%	73%	79%	68%	86%	91%
Modified attentional gist	100%	70%	75%	69%	83%	88%
Attentional gist	100%	53%	46%	41%	64%	60%

Table 4.3 shows the success accuracy when using these descriptors for the first six datasets. In all the datasets except *DS*-1, the success accuracy is highest when using *envelope gist* followed by *modified attentional gist*. The success accuracy is very low when considering *attentional gist*. In some datasets, it is below 50%.

A high success accuracy indicates a high contextual agreement, at least visually according to the image category criteria, between the test images and the images corresponding to the learned weight vectors. Intuitively, because the contextual agreement is high, so will be the weights (i.e., the optimum weight for the test image and the learned weight). On the contrary, low success accuracy (as when using *attentional gist*) indicates that the hetero-associative maps the contextual descriptor into a weight that is very different from the optimum weight.

The reason that *attentional gist* performs well in *DS*-1 is because this is the only dataset where the background remains the same across all the images and hence there is no visually discriminative contextual difference. This explains why we have 100% success rate for this dataset as all the images have the same label (see Table 4.3). For this reason, we can see that the detection performance of this descriptor is comparable to that achieved by the other two descriptors.

To complement the above results, a visual example is shown in Fig. 4.12(a). Two sample images one from *DS*-1 and the other belonging to *DS*-5 are selected for demonstration. Next to these two images, the optimum weight vectors learned through PSO are shown (indicated in the figure as ′optimized weight′). The predicted weight vectors by the hetero-associative model when using the three descriptors is indicated in the figure as ′Learned weight′. Note that for *DS*-5 sample image, the learned weight vectors exhibit high similarity to the optimized weight vector when using *envelope gist* and *modified attentional gist* descriptors and low similarity when using *attentional gist*.

The implication of the weight variation is shown in the saliency maps generated by these learned weights. For *DS*-5, the saliency maps produced by the learned weights from *envelope gist* and *modified attentional gist* descriptors exhibit high precision in detecting the target object (the red ketchup bottle) with few false negative regions. The saliency map generated by the *attentional gist* weights produces high number of false positives.

Additionally, the obtained learned weights from each descriptor are compared to all the optimum weight vectors used in training the hetero-associative model using Pearson correlation. Figure 4.12(b) shows the best correlation coefficient values and the corresponding training images. Note the contextual label similarity between the test image and the matched images through *envelope gist* and *modified attentional gist* descriptors. The label is different when using *attentional gist* descriptor for *DS*-5 sample image.

For *DS*-1 sample image, the learned weights from all the descriptors are highly similar to the optimum weight vector because all the images in this dataset have an almost constant background. This is also reflected by the generated saliency maps and the matched images (all having the same label as the test image).

From the detection accuracy performance shown in Fig. 4.11, the success accuracy result provided in Table 4.3 and the visual example in Fig. 4.12, it is obvious that the selection of a contextual descriptor plays a major role in learning an appropriate hetero-associative model for a correct weight vector prediction. Both *envelope gist* and *modified attentional gist* can be considered for this purpose. Our design selection prefers *modified attentional gist* over *envelope gist* as it is computationally more efficient than the former, despite having slightly less detection accuracy than that produced by *envelope gist* descriptor.

4.4.2 Target object feature weighting

A detailed analysis of the target based feature weighting is provided in this section. Since we want to focus on studying target objects and their relevant features, we eliminate any involvement of the intra-contextual variation between datasets. For this reason, only *DS*-7 is considered as all the six objects under study are contained within the images of this dataset.

For each of the six objects in *DS*-7, the best features are examined and how would they affect the generation of the target object weighted saliency map $SM_{TD}^{w_2}$. In addition, the descriptive power of the features used to characterize the target object is explored. We mean by descriptive power is the extent to which such low-level features can describe the target object. Furthermore, we would like to see whether this kind of weighting is able to detect the target object or not. Finally, it is important to assess the independence of the target features that establish the target object weight from the contextual content of an image and to which extent $\mathbf{w}_2$ is different from $\mathbf{w}_1$.

4.4.2.1 Target feature importance and interpretation

Figure 4.13 shows the target object weight $\mathbf{w}_2$ profile of the six objects in *DS*-7. The weights are averaged over all the train/test splits. There are

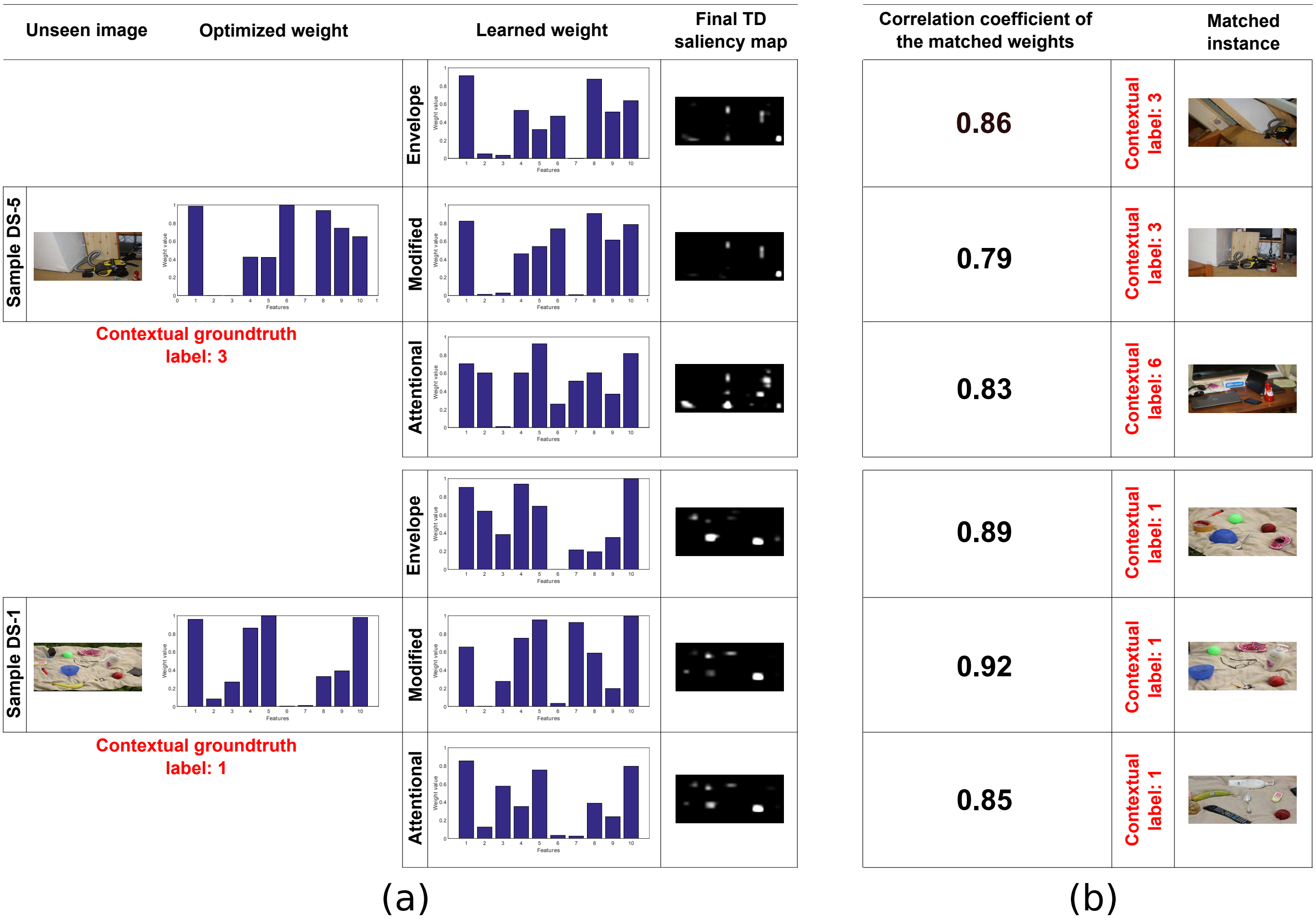


Figure 4.12: An example showing the impact of the contextual descriptor on the predicted weight vector by the hetero-associative model. a) two images are chosen, one from *DS*-1 and the other from *DS*-5. These two images are labelled contextually according to the criteria mentioned in section 4.4.1. The first image has a label of 3 and the second one is labelled 1. The 'optimized weight' column represents the optimized weight vectors that are learned through PSO. The 'Learned weight' column gives the predicted output weight vectors of the hetero-associative model when using the three descriptors. This is followed by the final top-down saliency maps generated through the corresponding learned weights. b) To access the learned weights from these descriptors, the learned weights are compared with all the optimized weight vectors of the training images using Pearson correlation. The image corresponds to the highest correlation result is shown. Compare the label of these matched images with that of the test images for each descriptor used in the process.

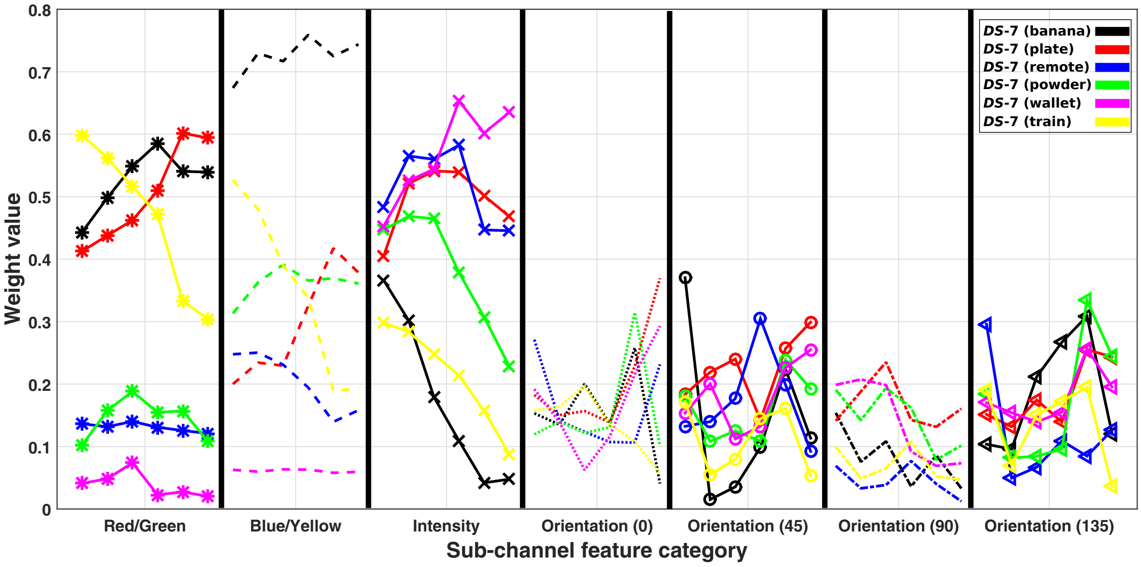


Figure 4.13: Target object feature weighting profile of *DS*-7 dataset. The plot is divided into seven partitions, each corresponds to the weight profile of a sub-channel feature at various scales. The x-axis in a partition represents the scale number whereas the y-axis is the actual weight value computed using Eq. (4.7). Only the average weight over the training splits is shown in the figure.

seven partitions in the figure where each split represents the weight profile of a sub-channel feature. Furthermore, there are six plots in each partition that corresponds to the weight profiles of the six objects. Each plot has six points such that each point on the plot is a single weight value of a sub-channel feature at a particular scale. The points correspond to the six scales are in order (i.e., scale-1,scale-2,...,scale-6). For instance, for *DS*-7(banana), the weight values of all the sub-channel features for the first scale is $[0.44, 0.67, 0.36, 0.15, 0.37, 0.15, 0.10]$ (see the first point value of the black plot in each of the seven partitions).

As different feature scales carry different information about the feature and would affect the mean and variance of the sub-channel feature, and potentially the weight value, the weight variation within scale for any sub-channel feature is obvious. In all our experiments, we use six scales as was originally proposed in the *Itti* model [1].

For any particular scale, it is easy to extract the important features based on the individual weight values. For instance, in *DS*-7(banana), only by considering the first scale, we can see that the *B/Y* feature is the most important and receives high weight value. This is expected as the yellow colour is the obvious feature of this object. Note that the *R/G* receives a moderate importance as the object does not exhibit a pure yellow colour and some mixture of other colours triggers the *R/G* channel. Interestingly, the orientation at 45° has higher weight values than the other orientation features. Similarly at 135°, 0.32 weight is assigned to this sub-channel feature at scale-5. These two features capture the geometric structure of the object because if we consider the angle of the two ends of a banana, they are approximately apart by 90^o. Hence, roughly, if the banana is positioned at any angle, one end will direct towards 45^o and the other toward 90^o.

Table 4.4 summarizes the three most important sub-channel features based on the maximum weight value at any scale for the objects in dataset *DS*-7. Hence the table provides the most descriptive features that characterize these target objects. The colour conspicuity is obviously the main driving feature that clearly captures some characteristics of the target features. Of course, this is true when the object has colour contrast features close to either *R/G* or *B/Y*. The importance of such features remains prominent when the objects' colour contrast does not match these two contrast features (e.g., the *DS*-7(wallet) object).

The intensity sub-channel is amongst the important features for five datasets. This feature although generates high weights on some objects, the feature does have the characteristic of being classified as a target feature. Because the intensity reflects pixel contrast, which in turn depends on the content of the image, intensity does not qualify to be a target feature. This can be seen in the intensity conspicuity map generated for two images from *DS*-7 where the object under study is the white power bottle (see Fig. 4.14). From the table, the intensity feature has the most importance (approximately 0.46 weight value as the maximum over scales (see

Table 4.4: Three most important features in *DS*-7 dataset for the six objects. The importance is based on the individual weight values assigned to these features when learning the target weight vector $\mathbf{w}_2$.

Feature rank	**Banana**	**Plate**	**Remote**	**Powder**	**Wallet**	**Train**
First feature	*B/Y*	*R/G*	*I*	*I*	*I*	*R/G*
Second feature	*R/G*	*I*	*Ori.* 45^o	*B/Y*	*Ori.* 0^o	*B/Y*
Third feature	*Ori.* 45^o	*B/Y*	*Ori.* 135^o	*Ori.* 135^o	*Ori.* 135^o	*I*

Fig. 4.13)).

For the first image, when the intensity contrast is high due to a dark background, the object is highlighted accurately, giving rise to high weight value on this features. However, when the same object is presented in a much brighter background, its saliency is low. This inconsistency is mainly attributed to background contrast with respect to the object and does not provide any descriptive influence on the target. Despite this behavior on the intensity feature, if the context and illumination conditions are kept the same in both training and testing phases (i.e., constrained visual condition), the feature can be associated with the target characteristics.

The orientation feature, has a more complex structure that depends on the geometry of the object. Although it takes small weight normally, it has high importance in certain objects (e.g., *DS*-7(banana) at 45^o and 135^o). However, this importance could be misleading in categorizing the feature as a target feature (e.g., *DS*-7(remote) at 45^o and 135^o).

To understand this contradictory behavior through an example, two images are selected to study the geometric structure of the two objects as shown in Fig. 4.15. To have a more detailed visual view on the feature

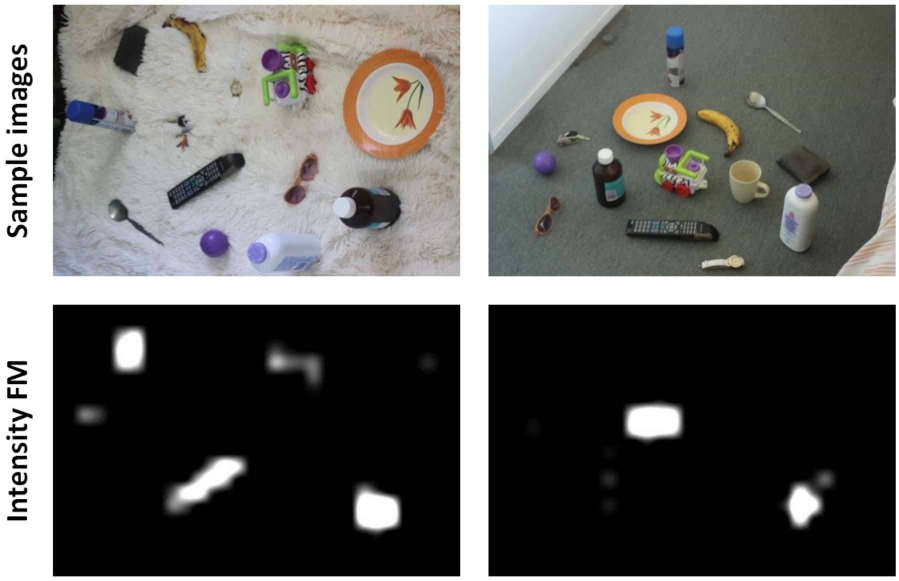


Figure 4.14: Sample images showing intensity variation due to background contrast variation.

response to these two images, the Gabor pyramid response is shown at 45° and 135° angles.

Note the consistency of the responses at both orientations in the case of the banana object (positioned at two different angles). The responses of both features capture the two ends of the object consistently (see the green circle in both images). The response to the remote control object is weak and inconsistent on the same images (indicated by the red rectangles).

In the second image, the response is very strong along the direction of 135°. This is due to the position of the object at this angle. For the same object, the response is very weak (almost negligible) at 45°, although Fig. 4.13 shows a good weight value on both features for this object. Similarly in the first image, when the object is nearly vertically positioned with reference to the angle of the acquisition device, the response of both features are insignificant. Hence, the orientation at these two angles to some extent represents target features for the banana object whereas for the remote control, they respond to object’s positional angle rather than being

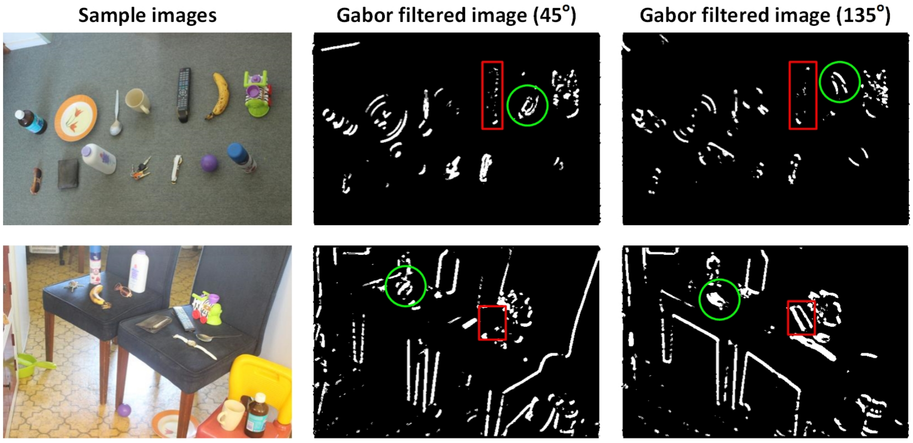


Figure 4.15: Sample images showing the extent to which orientation features can be considered as target object features. The green circle indicates the banana object while the red rectangle is for the remote control object.

true object features.

In summary, the results discussed above suggests that the target feature weighting is influenced by the low-level features of the object that vary in importance. The descriptive power of these low-level features varies accordingly. In most of the datasets discussed above, the colour feature is considered to be the most prominent features that would describe the target object amongst these low-level features. Intensity, is a contrast based feature and varies with the background illumination and hence, it is considered to be a week descriptive feature. Finally, the orientation has limited descriptive capability that varies with the geometric structure of the object itself.

4.4.2.2 Target object weighted saliency map $SM_{TD}^{w_2}$

From the weight profile discussed in the previous section, it is clear that the primitive *Itti* model features used for target object weighting have limited capability. Despite the limitation, this kind of weighting frequently samples locations from images that are likely to contain the target objects

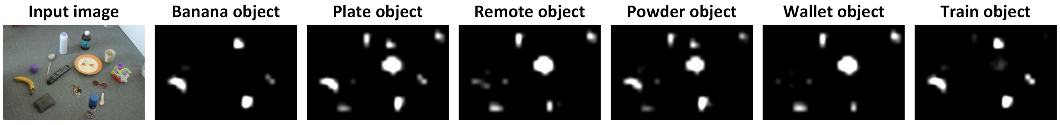


Figure 4.16: The descriptive nature of the learned features for the target object. Each map is generated by weighting the attentional features by the learned target object weight vector $\mathbf{w}_2$. The activation regions in each map explains the nature of the target features and their contribution in detecting the target object.

if the important features are weighted highly. Concretely, if a particular feature is weighted highly (e.g., *R/G*), then all the objects characterizing by the same feature will be highlighted along with the target object. For instance, in Fig. 4.16, when multiplying the features with weights that are learned for the banana object, and because one of the important target features describing this object is *B/Y*, the target object is detected but along with blue colour shades from two other objects. Similarly, the *R/G* feature has high weight values (see Fig. 4.13) and as a result, the wheel of the toy train is also highlighted albeit lower less saliency.

For the remote control object, the most important according to Table 4.4 represents the intensity feature whereas the orientation features are insignificant for the reason discussed earlier. Similarly as we hypothesized earlier, intensity features cannot be considered a target object feature unless we have constrained visual setup. This is reflected by the saliency map generated for the remote control object as shown in Fig. 4.16. The remote control does not exhibit a high intensity contrast due to the background as it is the case with the plate object, thus results in conspicuity decrease for this object as shown by a very small and weak activation region in the map. Similar reasoning and feature analysis can be performed with other objects in the figure.

In summary, based on the features used throughout the proposed model, these features are effective in detecting the target object only if they are being detected with high conspicuity and being further weighted appropriately by the important features. In addition, if a target feature has a high weight, it will generate false positives if other objects or regions share the same features.

4.4.2.3 Relation between $\mathbf{w}_1$ and $\mathbf{w}_2$ weights

We have seen that the target based feature weighting makes the best attempt to give importance to features that are better descriptive of the target object. In the case of $\mathbf{w}_1$, the optimization process performs similar tuning of weights but through an objective function that aims at giving weights to the most important features that are best suited for the candidate image only. Hence, it is very likely that the optimizer gives high weights to features that receive importance by the target weighting process. However, it is also possible that it tunes the weights differently to maximize the detection accuracy without attaining to the target object features.

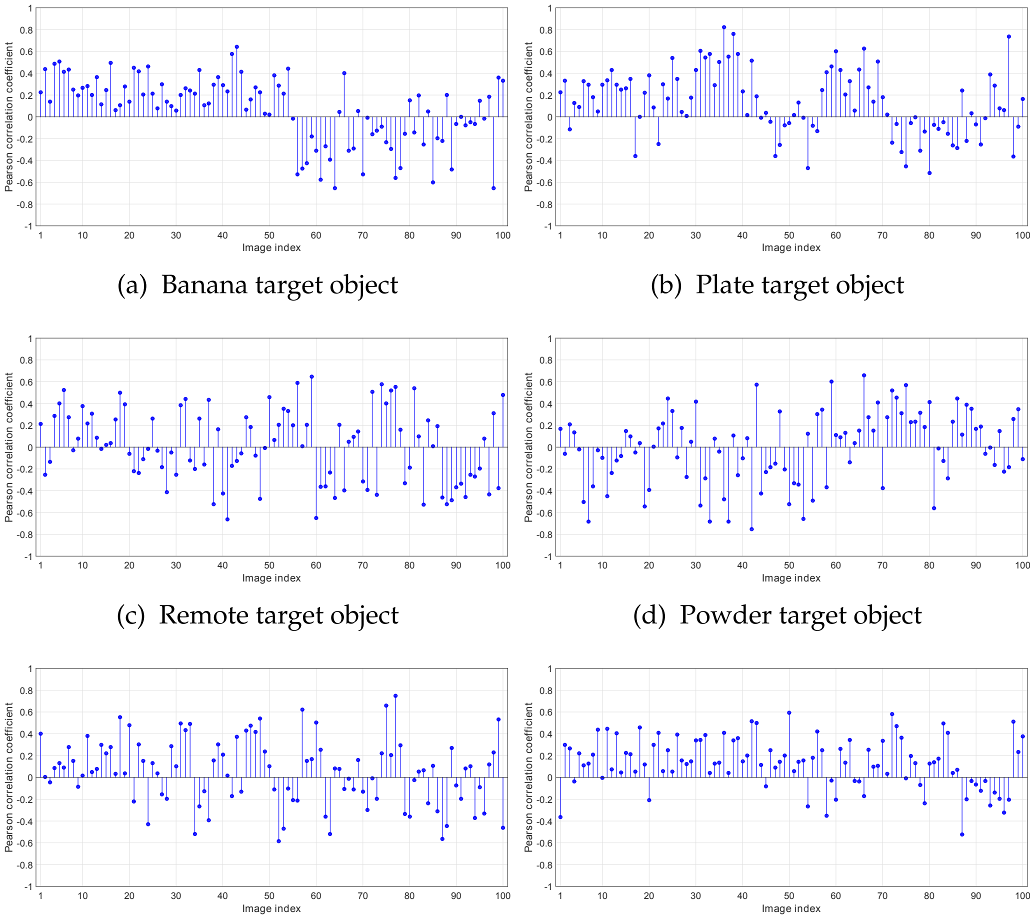
To emphasize this point, a correlation between the target object weight vector $\mathbf{w}_2$ and the optimized weights vector $\mathbf{w}_1$ is performed. For a fair comparison, the dimension of $\mathbf{w}_2$ is reduced to only 10 to match that of $\mathbf{w}_1$. This is done by first averaging the weight values obtained from all the train/test splits. Furthermore, for each sub-channel feature, the weights obtained for the six scales are averaged to produce a single weight for the corresponding sub-channel feature, and hence yielding a seven sub-channel feature weight vector. A similar averaging is performed over the CM feature weights and when concatenated with the sub-channel features weight, the weight vector length becomes 10.

Figure 4.17 shows the Pearson correlation coefficient between the target weight vector and the individual image optimized weight vector (total of 100 images in the *DS*-7 dataset). The correlation will give us evidence of the level of similarity between the two types of weights that we expect to be low. If both weights are highly correlated, then this means that the weights are redundant.

Note that the correlation coefficient is very low in many examples from the six target object correlation profiles, and in only few examples this coefficient exceeds 0.6. A low coefficient suggests that the weights from the optimization process tunes the weights differently from that performed by the target feature process. For all six datasets, the correlation profile provides enough evidence that the values of both weight vectors are different. The result justifies the use of two different types of weighting mechanisms as they capture different aspects of the detection process and they are not redundant.

To complement the correlation result obtained above, Fig. 4.18 shows six sample images that are selected from the wallet and train objects to give an insight on the values of these two types of weights. For each of these images, the corresponding saliency maps generated by $\mathbf{w}_1$ and $\mathbf{w}_2$ weight vectors are shown. The six examples are those that yielded the best, average and worst correlation coefficient values. Note that the $\mathbf{w}_1$ values vary in the six examples while $\mathbf{w}_2$ remain the same for the images belonging to a particular object type as it represents the target object features weight vector. In addition, the CM weights of $\mathbf{w}_2$ do not exactly follow the previously discussed procedure of assigning the CM weights through the maximum value over the sub-channel feature. Instead, the CM values are averaged over all the splits as with the sub-channel feature weights.

Starting with the worst correlation example for the train object, the corresponding correlation coefficient value is approximately -0.522 suggesting that the weight vectors are highly uncorrelated (see the weight vectors shown in Fig. 4.18 for the worst case train object image). The weight contribution on the individual sub-channel feature in $\mathbf{w}_2$ supports our previous observation regarding the important features for the train object (refer to Table 4.4 and Fig. 4.13). The *R/G*, *B/Y* and *I* received higher weights than the orientation features as expected. The feature that yields the target object detection in this example is the *R/G* (visually observe the red wheels of the train). If we look at the $\mathbf{w}_1$ weight vector, we can see different weight values on the sub-channel features as well as the CM features. The weight profile shows very high weights on the orientation and inten-



(a) Banana target object

(b) Plate target object

(c) Remote target object

(d) Powder target object

(e) Wallet target object

(f) Train target object

Figure 4.17: The Pearson correlation coefficient between the weight vectors $\mathbf{w}_1$ and $\mathbf{w}_2$ for all the individual images in the *DS*-7 dataset for all six target objects. The $\mathbf{w}_2$ weight vector is formatted by taking the average of sub-channel feature weights over the entire training splits to reduce its dimension to 10 to equal that of $\mathbf{w}_1$ for a fair correlation.

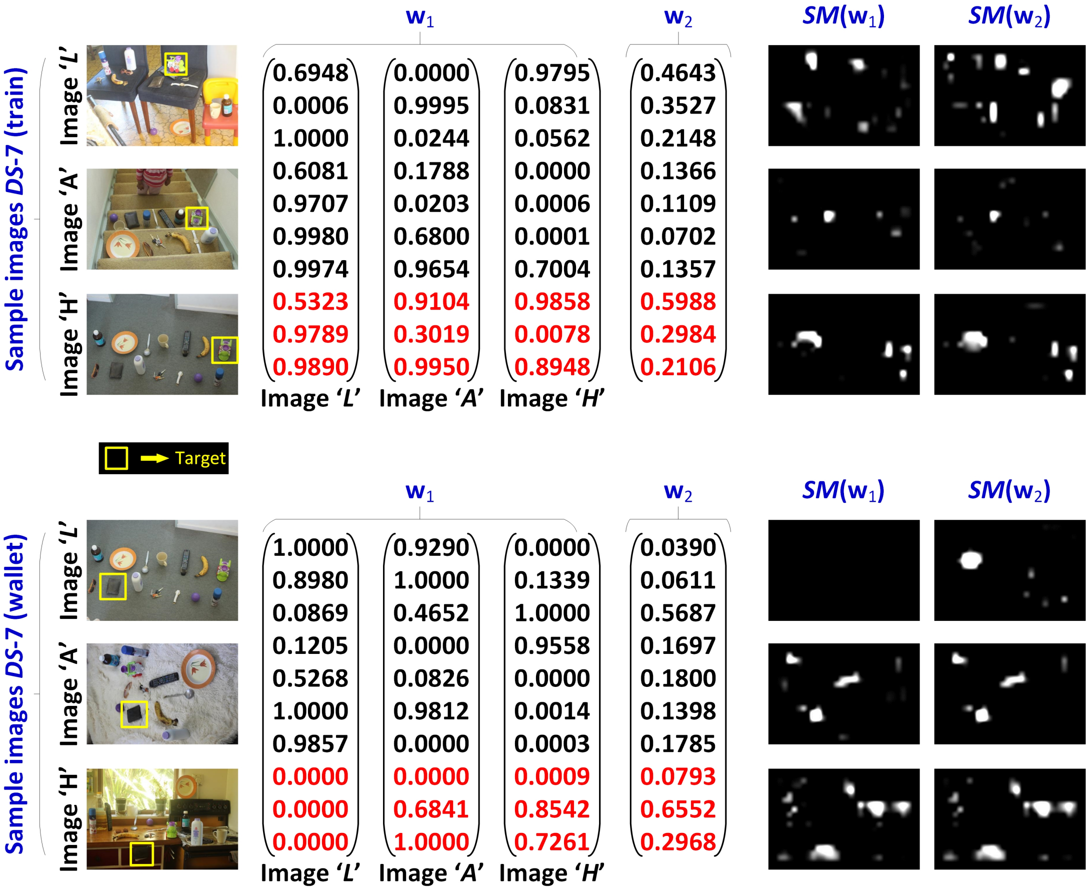


Figure 4.18: Sample images from the wallet and train objects showing the obtained weight vectors $\mathbf{w}_1$ and $\mathbf{w}_2$ with the corresponding saliency maps generated by these weights for the images having the lowest (abbreviated in the figure as $'L'$), average ($'A'$) and the highest ($'H'$) correlation coefficients between the two weight vectors. Note that for either object, there is only a single $\mathbf{w}_2$ vector the represents the target object feature weights. The last three weights in a vector shown in red correspond to the CM features weights. The saliency maps are generated using the respective weights.

sity sub-channel and CM features (more than 0.9). In contrast, both the colour sub-channels and their CM feature received lower weight with respect to those assigned to orientation features. The weight distribution is opposite to the weight distribution over the features when $\mathbf{w}_1$ is used.

The impact of this weight difference is very obvious in the saliency map generated by $\mathbf{w}_1$. Because the intensity feature receives a very high weight value, most of the high-intensity regions of the image are activated in the saliency map $SM_{TD}^{w_1}$ (e.g., the powder and the plate objects). Even for the target object itself, the dominant features in detection is the intensity rather than the colour. Since the objective was to maximize the detection accuracy only for the image under consideration, considerable variation in weight importance exists in $\mathbf{w}_1$. Hence, for this image, intensity yielded the highest detection accuracy even higher than the R/G feature that represents one of the target object features.

In another example, the weight vectors of the best-matched (with a correlation coefficient of 0.748) wallet object image are analyzed. Because of the high agreement between $\mathbf{w}_1$ and $\mathbf{w}_2$ weight values, the resultant saliency maps generated by the two weight vectors also exhibit high agreement in terms of the salient regions. As depicted from Table 4.4 and Fig. 4.13, the important features for this object are the intensity and orientations at 0° and 135°. This can also be noticed in the $\mathbf{w}_2$ weight vector of the best-matched image, as these features receive higher weight values than the colour sub-channel $\mathbf{w}_2$ features. A similar weight pattern is found in $\mathbf{w}_1$. The intensity feature receives the maximum possible weight value of one. Both colour sub-channel features receive low weights compared to the intensity weight. The orientation feature weight is very low except for the orientation at 0° which is slightly higher. Although the weight on the orientation CM is relatively high (0.726), it has no major effect on the final saliency map as already the importance is suppressed by the low sub-channel features' weights.

The remainder of the sample images can be analyzed in a similar manner as above. In one case (i.e., the worst correlation for the wallet object), the saliency map does not have any activation regions. Despite the sub-channel features having high weight values, all the CM weights are zero, which results in no salient regions being selected on the map. The saliency map generated by $\mathbf{w}_2$ produced a saliency map with many false positives but without detecting the target (precision is zero). The target features for this object were learned over many examples where the intensity along with some of the orientation features were chosen to be the best available features. However, these features failed to highlight the target object.

PSO was able to overcome this limitation during the optimization search. During the optimization process, no such features were found that could detect the target object. As a result, the optimum solution was achieved by assigning zero values to the CM weight. Although this weighting reduces the precision to zero, it avoids generating unnecessary false positive region as it is the case with $\mathbf{w}_2$ weights.

Typically, from the two proposed feature weighting approaches used in our model, one through optimization process (i.e., $\mathbf{w}_1$) and the other through target object feature extraction (i.e., $\mathbf{w}_2$), we can see a prominent difference between the two weight values and the important features. For the individual images on the datasets, very low correlation coefficients were noticed when correlation was performed between the two weight vectors. On some occasions, the weight obtained from the optimization process agreed with the target object weights. As a result, the generated saliency maps from both weightings approaches highlight regions of the image that represent different information. Even for the target object region, both weights highlight the region differently which could be beneficial when combining both maps as we will see in the coming sections.

4.4.2.4 Object recognition analysis

As mentioned previously in section 4.2.4, the RTF module is an optional step in recognizing the target object from the candidate salient regions determined by the other two modules. In this section, we briefly discuss the recognition accuracy of the module without comparing it with current state-of-the-art computer vision object recognition techniques. This is because the purpose of this module is to reduce the number of false positives that is introduced when combining the two saliency maps from the other two modules. One of the main advantages of the proposed recognition module is that it is very efficient as no extra computational overhead is needed. Unlike the state-of-the-art recognition methods that require very complex descriptors in order to perform recognition, the proposed module reuses the same basic features that are extracted from the image for saliency map generation (see section 4.2.4).

In section 4.2.4, we have described how views are extracted from a saliency map and based on these views, a Naive Bayes (NB) classifier is learned that classifies target and non-target views. A single view represents a vector of feature values of length 42. The location of a view on a saliency map is determined by the activation regions in that saliency map.

During the test phase, once the saliency map is generated by combining $SM_{TD}^{w_1}$ and $SM_{TD}^{w_2}$, the most salient regions are kept by performing an adaptive segmentation procedure typically applied on saliency maps (refer to section 2.2.4.3.3 for the mean-shift adaptive segmentation for saliency maps). All the resultant activation points on the segmented binary map are potential view locations that need to be classified by the NB classifier.

To evaluate the accuracy of the recognition module, two classification accuracies are computed. The first accuracy is determined at the *'views* level. This is equivalent to the number of views that are classified correctly (either target or non-target) with the help of the groundtruth map. The groundtruth map determines the actual locations on the image belonging to the target object. If TP_v represents the total number of true positives views (i.e., correctly labelled as target by the NB classifier), and TN_v as true negatives, then this accuracy is given as:

$$\text{View accuracy} = \frac{TP_v + TN_v}{\text{Total number of views}} \tag{4.16}$$

The second accuracy is computed at the salient region level. This accuracy is used to evaluate the performance of the majority ranking decision approach in eliminating false negatives. A salient region on a segmented binary saliency map is defined as the activation points that are 8-connected. Two activation points are said to be 8-connected when they are immediately adjacent to each other in any direction [145]. A region is labelled to be a target object (or to contain a target object) if it fully or partially coincides with the groundtruth according to a hit criteria.

The hit criteria is defined by two threshold values (T_1 and T_2). If $|R^u|$ represents the total number of activation points in a segmented binary salient region u, and $|\hat{R}|$ is the number of pixels of the target object, then the hit criteria is defined as follows:

$$\mathcal{L}_u = \begin{cases} 1, & \text{if } \left(\frac{\text{TP}_u}{|\hat{R}|} \geq T_1\right) \text{ and } \left(\frac{\text{FP}_u}{|R^u|} \leq T_2\right) \\ 0, & \text{otherwise} \end{cases} \tag{4.17}$$

where $\mathcal{L}_u$ is the label of the salient region u. TP_u represents the overlapped region between the groundtruth map and the salient region u whereas FP_u is the non-overlapped region between them. Hence, the above procedure for labelling regions in the saliency map yields region based groundtruth labels for a saliency map.

This hit criteria approach labels the region as negative if it coincides with the target object region, but should not have a large false positive region with respect to the size of the region itself. Also, if the region's false positive is small, but its TP region is very small with respect to the actual size of the target object ($|\hat{R}|$), then it does not pass the hit criteria and will be assigned a negative (i.e., non-target) label. For this reason, T_2 is chosen to be 0.5 to pass the first condition and T_1 is kept low (in all the simulations, it is kept at $= 0.1$) to fulfill the second condition.

When majority ranking is applied to the salient regions, a region takes the label of the majority class label of the *views* within that region. We refer to this label as region based predicted labels and it is denoted as $\widehat{\mathcal{L}}_u$. So the region based accuracy becomes:

$$\text{Region accuracy} = \frac{TP_{\mathcal{L}} + TN_{\widehat{\mathcal{L}}}}{\text{Total number of salient regions}} \tag{4.18}$$

where $\mathcal{L}$ and $\widehat{\mathcal{L}}$ are the region based groundtruth and the predicted label vectors for all the salient regions of the entire test images respectively. Note that the region based classification accuracy given by Eq. (4.18) is computed over all the candidate regions from the entire test image set across all the splits.

Figure 4.19 shows two types of confusion matrices along with the classification accuracy for nine datasets, *DS*-1, *DS*-2, *DS*-4 and the six objects from *DS*-7. The values in the confusion matrix in order from top-left to bottom-right represent True positive rate (TPR), false negative rate (FNR), false positive rate (FPR) and true negative rate (TNR) respectively.

The view based results show how well the classifier and the recognition features determine the class label of a single view from any location of the image (either belonging to a target or non-target). The region based recognition result depends on the view based result as the latter imposes an additional decision about whether a collection of predicted views belonging to an attended region. It is important to understand that the majority ranking decision dictates the final output by eliminating the false negatives, while the purpose of providing view based recognition results is to give a sense on the effectiveness of the classification approach and the proposed recognition features.

From both sets of confusion matrices shown in Fig. 4.19 (where the left of the figure gives the view based results while the regions based results are on the right), we can observe that TPR is normally very high except for the train object (bottom right in the figure) in which the TPR is lower than that in the other datasets. The TNR is good but in some datasets, the

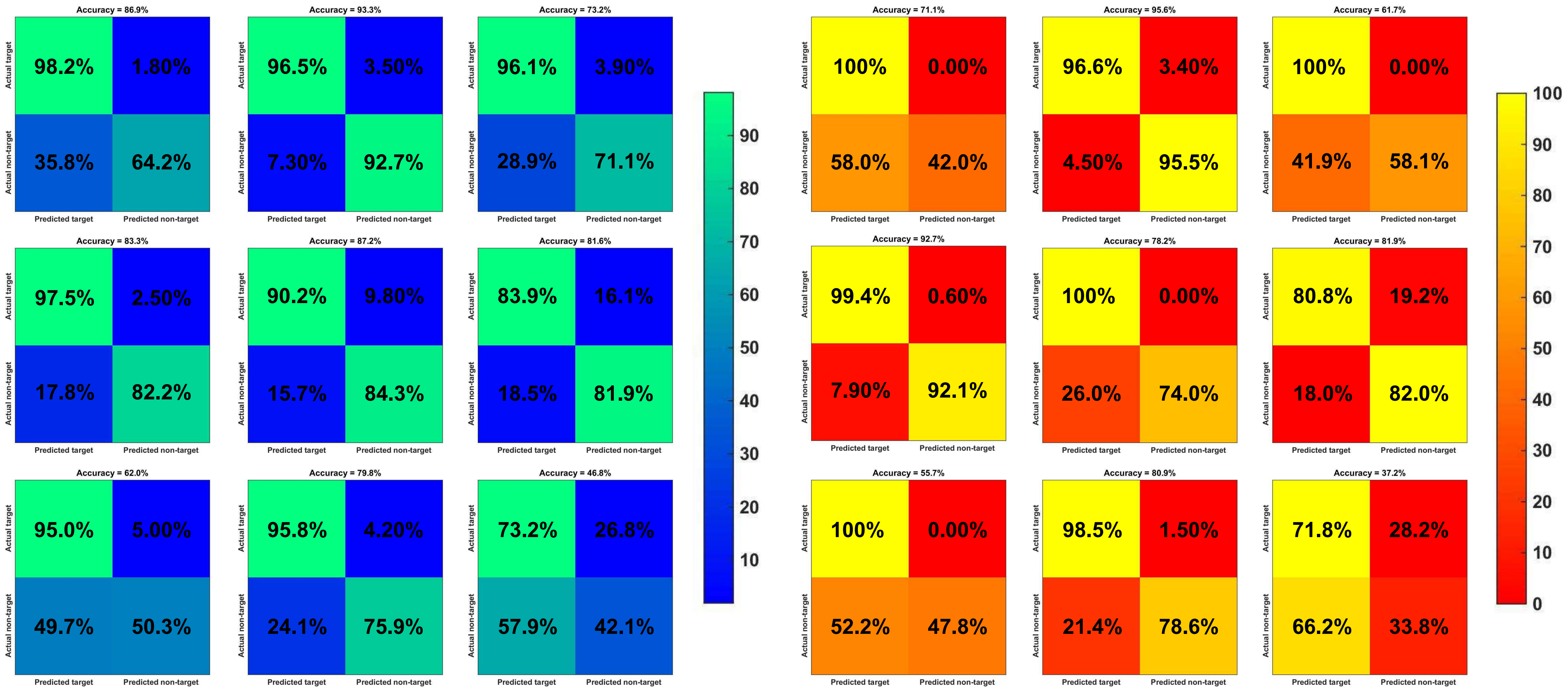


Figure 4.19: The confusion matrix for view based recognition on the left and attended region based recognition on the right. The view classification accuracy is calculated using Eq. (4.16) and region based accuracy using Eq. (4.18). The confusion matrix is constructed for the following datasets: (top row, left to right: *DS*-1, *DS*-2 and *DS*-4), (middle row, left to right: *DS*-7(banana), *DS*-7(plate) and *DS*-7(remote)), (bottom row, left to right: *DS*-7(powder), *DS*-7(wallet) and *DS*-7(train)). Each confusion matrix computes the true positive rate (TPR), false negative rate (FNR), false positive rate (FPR) and true negative rate (TNR).

performance is poor (e.g., *DS*-7(train) and *DS*-7(powder)). Furthermore, because the majority ranking decision is dependent on the accuracy of the view based recognition, its performance follows that of the view-based approach. In some occasions, the overall classification accuracy is boosted when majority ranking is considered (e.g., dataset *DS*-2). This implies that the majority decision-making approach provides a higher possibility in eliminating false negative regions even if many of they views are classified incorrectly.

Where the TNR is low in the view based recognition (e.g., in *DS*-7(powder)), it implies that the recognition feature space for the two classes is complex and not well separable. This could be attributed to the object itself as it might not be possible to describe the target object using the basic set of features used here, hence making it difficult to distinguish from other non-target regions.

For the powder object specifically, the classifier yields high TPR values. This is because one of the most important features for this object is the intensity. The classifier was able to classify most of the positive views belonging to the target object and exhibiting high-intensity values correctly. Many other views that are sampled from various regions of the dataset are also being classified positively as these views have similar feature profile, at least in intensity, to those being classified positively. As a result, the FPR is high, resulting in classification accuracy degradation.

When the majority ranking decision is employed, in some datasets the TPR reaches the theoretical maximum (i.e., 100%). This implies that if a saliency map detects a target object along with some false positives, it is guaranteed that the classifier would not eliminate the correctly detected regions. As a result, the detection accuracy would never be degraded by the RTF module. Either it will be improved by eliminating false negatives or remain the same. When TPR is less than 100%, it suggests that in some examples, the positively detected regions are eliminated incorrectly by the classifier. However, as we will see in the detection accuracy section that follows, in practice the overall performance over all the splits and examples are improved by the classifier.

4.4.2.5 Detection accuracy performance

The final output of all the modules is a top-down saliency map SM_{TD} that is intended to highlight the target object with high likelihood. In order to evaluate the detection accuracy performance, the *F*-measure is computed by calculating the average precision and recall values between the top-down saliency map and the groundtruth map at different scales over all the test examples in a particular split. Table 4.5 summarizes the obtained *F*-measure scores in all 12 datasets and for various saliency maps including the final saliency output SM_{TD}. Note that the recognition is performed on the combined map (either the arithmetic or *AND* based) that has higher mean *F*-measure value. In addition, all the results presented in this table are based on using the proposed *modified attentional gist* contextual descriptor.

In many datasets, particularly those associated with *DS*-7, the mean *F*-measure is very low (e.g., *DS*-7(plate) and *DS*-4). A low detection accuracy suggests that the basic attention features used throughout the three modules are insufficient for detecting the target objects contained in these datasets. On other datasets, e.g., *DS*-1, the performance is high because the target object (red cricket ball) is well characterized by the *R/G sub-channel* features used in the model.

Typically, the maps generated through the contextual optimization weighting $\mathbf{w}_1$ have higher detection accuracy compared to those generated by $\mathbf{w}_2$. Also, the combination strategy based on *AND* operation yields higher *F*-measure values than those produced by the arithmetic mean approach. Furthermore, whenever the recognition process is applied to the combined map, the performance is always improved by either using the majority ranking or the weighted approaches.

To have in-depth analysis of the results in Table 4.5, we analyze *DS*-

Table 4.5: The target detection accuracy performance of the proposed model using *F*-measure score. The performance of the final top-down saliency map output of the model is indicated in last two columns for either majority or weighted recognition approaches. Performances of the individual modules of the model are also presented for comparison along with the performance when only BU *Itti* model is used. The saliency map that gives the best accuracy performance for each dataset is highlighted in red.

Dataset	**BU**	$\mathbf{w}_1$	$\mathbf{w}_2$	**Arith.**	***AND***	**Best**	**Majority**	**Weighted**
DS-1	0.33	0.56	0.48	0.55	0.71	*AND*	0.77	0.73
DS-2	0.06	0.12	0.10	0.13	0.18	*AND*	0.30	**0.31**
DS-3	0.04	0.14	0.13	0.16	0.19	*AND*	0.29	0.29
DS-4	0.02	0.03	0.06	0.05	0.05	*AND*	0.0983	0.10
DS-5	0.15	0.36	0.33	0.35	0.41	*AND*	0.55	0.50
DS-6	0.24	0.41	0.31	0.38	0.45	*AND*	0.2967	0.42
DS-7**(banana)**	0.02	0.08	0.06	0.07	0.06	Arith.	0.24	0.23
DS-7**(plate)**	0.23	0.36	0.25	0.30	0.31	*AND*	0.42	0.40
DS-7**(remote)**	0.02	0.05	0.04	0.05	0.04	Arith.	0.10	0.08
DS-7**(powder)**	0.05	0.12	0.07	0.10	0.13	*AND*	0.17	**0.19**
DS-7**(wallet)**	0.05	0.08	0.08	0.08	0.09	*AND*	0.19	0.17
DS-7**(train)**	0.03	0.15	0.09	0.13	0.13	*AND*	0.13	0.13

1 dataset starting from the bottom-up (BU) accuracy performance. Note that the bottom-up Itti process generates the saliency map by assigning uniform weights to all the sub-channel features as well as the CM maps. The bottom-up saliency gives a good mean *F*-measure of 0.33 because the target objects are mostly salient in this dataset with backgrounds that are almost uniform without clutter.

A considerable boost in performance is achieved when $\mathbf{w}_2$ weights are used. An even better performance can be seen when the contextually optimized weights $\mathbf{w}_1$ are used. When the maps generated from these two weighting approaches are combined with the arithmetic mean approach, the performance is slightly degraded from that achieved by $\mathbf{w}_1$ weights alone. However, when the *AND* combination was used, a substantial improvement to 0.71 was achieved.

Normally, if the *F*-measure score by the arithmetic mean approach is lower than either of that from $\mathbf{w}_1$ or $\mathbf{w}_2$ weighting, it is likely to be that the false positives produced by the two weighting techniques are different, thus resulting in an increase in the false positives in the combined map. For *AND* based combination strategy, diversity in false positive regions does not degrade the performance as they will be eliminated in the combined map. Furthermore, improvement through the *AND* combination strategy also suggests a high agreement between the two maps in detecting the target object (i.e., high precision in both $\mathbf{w}_1$ and $\mathbf{w}_2$ generated maps).

When the recognition process is applied to the combined map through *AND* operation, the majority ranking approach was able to slightly improve the final performance. This is expected as the FPR is high as seen in the region based confusion matrix for this dataset shown in Fig. 4.19. As a result, it is possible that any false positive regions that are present in the combined map are classified as positive regions by the classifier and hence marginally affect the detection performance.

In *DS*-7(powder), the detection performance profile given in Table 4.5

is similar to that of *DS*-1 except that the detection accuracy is low in each type of map. Another noticeable difference is the weighting decision performs better than the majority ranking recognition decision (*F*-measure score of 0.17 in the latter and 0.19 in the former). Table 4.6 shows some statistical information on the weight values of both datasets to clarify this difference.

In *DS*-1, the mean weight on the candidate salient regions that correspond to target object is 0.86 and 0.51 on non-target region. Approximately 60% of the total non-target regions have weights greater than 0.5 (see the number in parenthesis). This means that the majority of the weights of the non-target regions are close to the target weights. In contrast, in *DS*-7(powder), the average weight on the target regions is high. The average weight on non-target region is low and further apart from the target weight. The further these two weights are apart, the closer would be the performance of the weighted approach to the majority ranking or even better as it is the case in *DS*-7(powder).

Table 4.6: Recognition weighting information for two datasets. The number of regions is the total number of candidate regions generated by the combined saliency map through the *AND* operation. The candidate regions correspond to either the target object or to non-target regions (i.e., false positive regions). The mean and the standard deviation of the recognition weight values over all images from all the splits for both regions are given. In addition, in the number of regions column, when the regions belong to false positive, the two numbers between the parenthesis correspond to the number of regions that have weights greater than or less than 0.5 respectively.

Region type	***DS*-1**			***DS*-7(powder)**		
	No. of regions	Mean weight	Std. weight	No. of regions	Mean weight	Std. weight
Target	$1,250$	0.86	0.08	262	0.94	0.10
Non-target (FP)	$1,242$ (731,511)	0.51	0.37	$1,172$ (563,609)	0.45	0.41

In dataset *DS*-7(train) (see Table 4.5), we find that the best performance is achieved by the optimized weight vector $\mathbf{w}_1$ before applying the recognition procedure. This performance is better than even when either combination strategies is applied. Since the recognition is applied to the combination strategy that yields the best performance (in this case the *AND* approach) when either the majority or weighted recognition approaches is used, the performance does improve over that achieved by the combination approach but does not improve beyond $\mathbf{w}_1$ performance. This suggests that the performance of the recognizer is bound to the quality of the saliency map to which the recognition process is applied. Note that because this object (train) comprises multiple parts with different shapes and colours, both $\mathbf{w}_1$ and $\mathbf{w}_2$ highlight different parts of the target region and yield diversity within the generated maps as hypothesized in section 4.4.2.3. Because of this diversity, it is very likely that the *AND* based combination strategy removes the candidate target object region from the combined map. If the candidate region is not available for the recognizer, its function would be limited to only classifying the false negative regions.

As a final example, note that in dataset *DS*-6, the best detection performance is achieved by the *AND* combination strategy, which yields an *F*-measure of 0.4493. When the recognition was applied, there was a considerable degradation in performance which is unlike any other example. This example suggests that the classifier has highly misclassified both the non-target and target regions. The TPR and the TNR for this dataset (not included in the confusion matrix figure) are 54.6% and 59.6% respectively. This poor classification is consistent with the observation we made earlier regarding the complexity of the train object.

To conclude the observations made in this section, typically, when all the modules are combined, the detection accuracy of the target object is boosted. The detection accuracy depends on the complexity of the target object and the extent to which the basic attentional features are able to describe such target objects. The optimal combination strategy and the recognition decision making are design choices that vary from dataset to another. The main focus of the results presented in this section was to show the importance of target object features in modelling top-down saliency. When the contextual information is combined with target object information, the detection accuracy is improved over applying BU process or when using only the contextual information.

4.4.2.6 Qualitative detection accuracy

Figure 4.20 shows the qualitative saliency maps produced by the proposed model which includes the maps generated by each module and the final top-down saliency map produced by the model. The sample images include one representative image from each dataset. The final three rows of the figure represent the segmented candidate regions that are highlighted by the groundtruth map (denoted as Seg_1), the weighted based recognition map (Seg_2) and the majority ranking based recognition map (Seg_3). The saliency maps generated by the arithmetic mean combination approach, the weighted recognition and the majority ranking recognition are abbreviate in the figure as *Arith.*, Rec_W and Rec_R respectively.

From these images, we can deduce that the proposed model on many occasions was able to detect the target objects with high precision and with very low FPR. For instance, the pink ball in *DS*-2 sample image was detected by the weighting and the majority ranking with a very high precision. The maps generated by $\mathbf{w}_1$ and $\mathbf{w}_2$ weighting schemes lack accuracy despite the target object being detected. This is because there are too many false positive regions. When the arithmetic mean combination strategy is used, many false positives still remained in the map. On the other hand, when the *AND* operation was applied, the false positive region reduced considerably. The recognizer was able to classify the one remaining false positive region correctly, resulting in high precision target object detection (compare the last three segmented images for this instance).

In the first image chosen from *DS*-1, the cricket ball is quite salient and the background is uniform. Despite this being a simple detection example, the final maps generated by both the majority ranking and the weighting approaches were unable to detect the target object with perfection. Along with the cricket ball, a false negative region sharing a similar *R/G* contrast profile to that of the target cricket ball has also being detected. The recognizer misclassified the region as the target.

The above example explains why the FPR for this dataset indicated in the confusion matrix is high. The majority of the false positive regions generated in this dataset share high feature similarity to the target object. These regions are classified incorrectly by the recognition module.

In another example from DS-6, the weighting recognition performed better than the majority ranking approach, showing the effectiveness of the weighting approach over majority ranking. The classifier misclassified the false positive region as well as majority of the views from the candidate target region that were highlighted by the *AND* combination map. As a result the majority decision removed the candidate target region from the final saliency map resulting in accuracy degradation. The weighting decision approach assigned a weight value of less than 0.5 to the candidate target region but did not eliminate the region totally.

From the above three example and the remainder of the images, We can visually confirm that the effectiveness of our model in detecting the target object. The proposed target and contextual weighting approaches when combined maximizes the detection accuracy even in complex scenarios with cluttered background and in the presence of distracting objects. The recognition model was able to eliminate the false positives successfully and sometimes failed to do so as can be seen in some of the examples.

4.4.2.7 Efficiency of the proposed model

In order to generate the final top-down saliency map of a new test image, it has to go through six computational steps. For a typical image having a resolution of 519×346, the approximate processing time for each of these

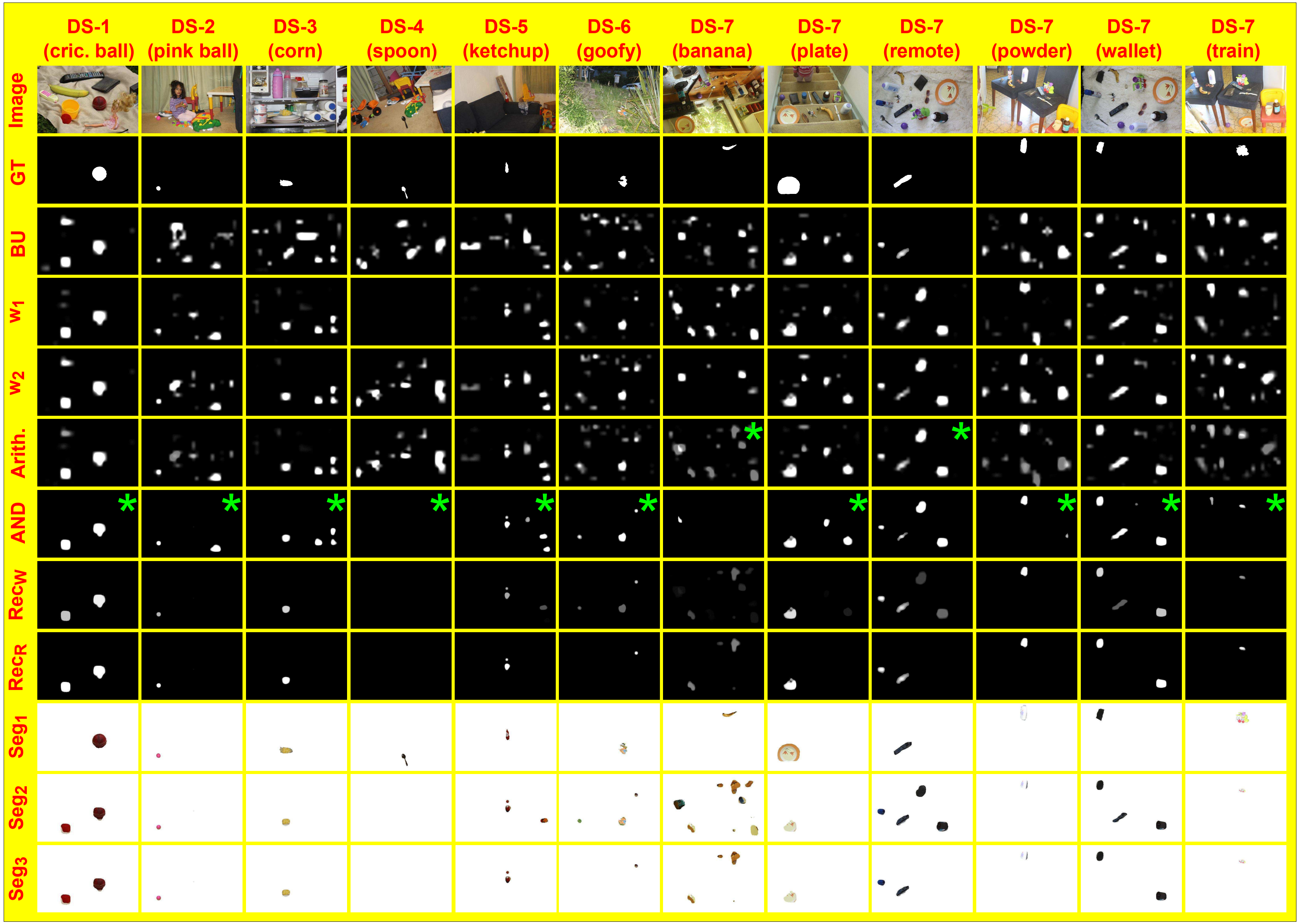


Figure 4.20: Qualitative analysis of the proposed model. The figure shows saliency maps generated by various modules of the model. The final generated saliency map represent the output of the recognition process using either the weighting approach (denoted as Rec_W) or the majority ranking (denoted as Rec_R) approach. The recognition is applied on the combination map that yields the best overall result on the corresponding dataset. The best combination strategy for the dataset is indicated by a green $*$. There is a total of 12 sample images, one drawn from each dataset. For a better visualization of the detection accuracy of the proposed model, the final three rows of the figure represents the segmented candidate regions that are highlighted by the groundtruth map (used as a reference for comparison and denoted as Seg_1), the weighted based recognition map (Seg_2) and the majority ranking based recognition map (Seg_3).

steps is as follows:

- (0.08 **seconds**) to extract and store the seven Itti attentional features at all scales.

- (0.01 **seconds**) to construct the contextual descriptor of the image. The processing time is given when the *modified attentional gist* descriptor is constructed. When *envelop gist* is used, around 0.7 seconds is required. The reason for the low processing time when *modified attentional gist* is used is because it simply utilizes the attention features extracted from the first step to constructing the descriptor and no additional feature extraction is required.

- (0.004 **seconds**) to predict the optimized weight vector $\mathbf{w}_1$ from the contextual descriptor using the hetero-associative model. This process is also very efficient as it simply computes the output weight through a linear combination of the contextual information as discussed in section 4.2.1.3.

- (0.006 **seconds**) to generate either of the two saliency maps $SM_{TD}^{w_1}$ $SM_{TD}^{w_2}$ from the optimized weight vector $\mathbf{w}_1$ and $\mathbf{w}_2$ respectively. Note that the target weight does not need to be predicted as $\mathbf{w}_1$ because it is directly loaded from the learned target features. Whether $\mathbf{w}_1$ or $\mathbf{w}_2$ is used to generate the saliency map, they are simply multiplied by the previously computed attentional features. Hence, the processing time is low and identical for either of the two saliency map generation processes.

- (0.0001 **seconds**) to combine the two maps using either of the approaches discussed earlier.

- (0.001,0.16 **seconds**) the first time represents the time required to predict a label of a single *view* from a candidate region by the recognition module. The second time represents the average time required to process a single saliency map with all its *views* across all the candidate regions.

From the above processing time profile, we can clearly see that only the recognition process takes most of the time while the other modules are very efficient. For an image dimension of 519×346 the total average processing time to produce the final saliency map is approximately 0.26 seconds. All the experiments are conducted on a single *Intel core i7-4790 @ 3.60GHz* machine with 8Gb memory and running on Linux operating system. The simulations are performed using MATLAB R2016b.

4.5 Chapter summary

The main objective of this chapter was to show whether the target object information can be useful when modelling top-down saliency. Furthermore, how to incorporate target information to the saliency model establishes a key element in modelling top-down saliency.

In this chapter, a novel top-down saliency model is proposed that combines both contextual and target information of the image. The model is based on attentional modules that learn two separate weight vectors which are tuned by the contextual information and the target object information. The model incorporates target information in two stages, one through target specific feature weighting and the other through a Naive Bayes recognition model. Hence, the proposed model consists of three modules, one for incorporating contextual information and the other two for target related information.

The model was tested and analyzed on seven challenging datasets with 12 different objects contained in complex and cluttered background scenes. The detection accuracy in the form of *F*-measure score is always boosted when incorporating target information, either through target feature weighting or through the proposed target recognition module. In addition, the proposed model is flexible as each of the three modules can be used separately for detecting the target object. However, the best detection is gained when all three modules are combined.

The proposed model only utilizes low-level features for detecting the target object. In all three modules, basic attentional low-level features comprising of colour, intensity and orientation are used. With this simple set of features, our model was able to detect complex objects with high precision. In some occasions, when the target object is very complex (e.g., part-based object), the proposed model fails to detect the object. This establishes a limitation on the proposed model as the low-level features provide less description of certain complex target objects.

The proposed approach represents a generalized model for a generic target object detection. Although for some complex objects, the detection accuracy decreases, this could be overcome by introducing more features into the model. Finally, the proposed model is very efficient. For an image of resolution 519×346, it only takes 0.26 seconds to generate the final saliency map that would highlight regions of the image likely to contain the target object. Hence, the model is suitable for real-time active vision applications.

Chapter 5

Modelling Visual Attention Combination Through Feature Selection

5.1 Chapter introduction and motivations

As mentioned in chapter 2, both theoretical and computational studies on visual attention provide significant understanding on the nature and functionality of top-down (task-driven) and bottom-up (data-driven) factors. However, as highlighted in section 2.3.2.3, the modalities and contribution of the two influences become less obvious when combined. This is typically the case in target object detection where it is not clear how the bottom-up influence complement the top-down influence. Some theoretical studies on human visual attention system also rule out the involvement of bottom-up influence in such high level tasks, and it is merely the top-down influence that dictates the process.

In computational literature on combining both influences, the most widely used approach is to combine them statically [23, 24, 28]. Hence, regardless of the content of the scene and for any target detection task, both saliency maps generated by the two processes are computed and combined in some way to yield the final saliency map. Two major drawbacks exist in combining both saliencies statically. First, in many cases the bottom-up saliency would not be effective in detecting the target object and might generate many false positive regions. Thus when combined with the top-down saliency, the accuracy would degrade. Secondly, if the first situation is true, then computing bottom-up saliency establishes a computational overhead and reduces the efficiency of the system.

It is well established that bottom-up saliency detects the most salient regions of an image. However, it is not always the case that the target objects are within these salient regions. In images where the saliency attributes coincide with that of the target object or when the target object lies within the sampling regions generated by a bottom-up process, the effectiveness of the bottom-up process in task-driven scenarios becomes more prominent. Furthermore, typically the bottom-up process has an edge over top-down because of the processing speed while the latter is more accurate in detecting the target object while performing a guided search.

Hence, motivated by the limitations of the current combination approaches and by the nature of both the influences, there is a need to devise a mechanism that would combine the two influences dynamically to maximize the detection accuracy while minimizing the computational overhead.

In this chapter, we formulate the combination problem as a features selection process that dynamically selects either top-down saliency, bottom-up saliency or both depending on the scene content. Modelling the combination process through feature selection will ensure a fair contribution of both top-down and bottom-up processes in detecting the target object effectively and efficiently. Despite the simplicity nature of our proposed approach in solving the combination problem, it can be generalized to any number of bottom-up and top-down saliencies. In addition, the proposed approach provides a way to understand and analyze how and when the bottom-up saliency positively participates in detecting the target object. Finally, our approach establishes a gateway for building more sophisticated and dynamic visual attention systems for target object detection.

5.1.1 Chapter objectives and overview

The main objective of this chapter is to propose a model that combines both top-down and bottom-up processes dynamically to maximize the detection accuracy of the target object and to improve the efficiency of the system. This is performed by proposing a mechanism to select either top-down saliency, bottom-up saliency or both depending on the scenario. Because any of these two processes consists of multiple features that contribute in generating the final saliency map (e.g., the Itti model), our selection is performed over the features rather than the final saliency maps. Thus, multiple features form bottom-up and top-down saliencies compete for the selection. Such a setup provides a better flexibility in combining both processes. For instance, if the selection is over the saliency maps, and as an example, if the bottom-up saliency is not selected, then all the features that contribute in generating the bottom-up saliency map will not be selected as well. It might be possible that certain features of the bottom-up saliency when combined with certain features of the top-down saliency would give the best solution. However, selection over saliency maps in this case would not yield an optimum solution.

For the above reason, our model formulates the problem as feature selection problem rather than saliency map selection problem for combining saliencies. The proposed model referred to as Feature Selection based Top-down Saliency Model (*FS-TDSM*) dynamically selects an optimized set of features belonging to either top-down, bottom-up saliencies or both. Our model follows the Itti model structure. The only difference between our structure and the Itti model is in the number and type of features being used. We explicitly include features that directly describe the visual properties of the target object which we refer to as top-down features. More about the nature of the top-down and bottom-up features are discussed in section 5.2.1.

To study the effectiveness of our model, we used five datasets. The target object to be searched for in each dataset represent a red cricket ball. Each dataset has specific visual characteristics (see section 5.3.1 for details). Hence, our feature selection model would select the best set of features that would maximize the detection accuracy of the target cricket ball for that dataset. Hence, a limitation of the current model is that it works within similar type of visually correlated images (i.e., in terms of background, illumination, clutterness, etc.). For each dataset, we provide a comprehensive study on the selected features and the contribution of the bottom-up and top-down saliencies in detecting the target object.

Note that in this chapter, we only demonstrated the effectiveness of our model on a single target (i.e., the cricket ball) and limited number of features. We have used the roundness and redness as top-down features as to some extent they describe the visual characteristic of the target cricket ball. Although we could have used more dynamic ways of extracting the target object features (e.g, using convolutional neural networks (CNN)) but that does not constitute the objective of this chapter. Hence, given a set of features that either belong to top-down or bottom-up saliencies (the number of features and the way they are produced are not important), our approach should be able to combine them using the set of best selected features from both saliencies.

The following steps provide a working summary of our proposed model:

1. Split the dataset under study into training and testing sets.

2. Apply the feature selection process using binary particle swarm optimization (BPSO) technique on the training set. After running the BPSO, the final product is a set of selected features having the best detection performance.

3. Repeat the BPSO selection process number of times (each implementation is called a run) on the same training set but with different seed initialization for repeatability.

4. Each BPSO feature selection run will produce a set of best selected features along with its achieved detection accuracy. Select the set of features yielding the best detection accuracy as the final solution. If more than one solutions exist (i.e., in case of equal detection performance), select the solution that comprises of minimum number of features.

5. Apply the selected features from the previous step on the images of the test set.

The following steps provide a brief overview of the BPSO feature selection process:

1. The objective function of the feature selection process is to maximize the *F*-measure score, which represent the detection accuracy of the target object.

2. Each particle is encoded by a *D*-bit where 1 indicates selection of a feature and 0 for a non-selection. A total of 15 features are used in the proposed work.

3. Once a particle's encoded in determined, its fitness value is computed. This is done by generating the saliency map only using the selected features through Itti framework. This is followed by comparing the saliency map with the groundtruth map to yield the *F*-measure score.

4. Step 4 is repeated for each image in the training set resulting in *F*-measure scores for each image in the training set for one particle. The *F*-measure scores are averaged to yield the final fitness value of that particle.

5. Step 5 is repeated for all the particles and constitutes a single iteration of BPSO.

6. The optimization is performed over number of fixed iterations (referred to as maximum number of iterations). This establishes a single run of the feature selection process. By the end of a run, the D-bit code of the particle having the highest fitness value is selected as the best feature combination for that run.

5.2 Feature selection based top-down saliency model (*FS-TDSM*)

The proposed model works in two phases, feature extraction and feature selection. Once the features are selected in the training step, they are used to generate saliency maps of a given novel image. The model tries to ensure that the selected features can lead to an optimized solution that would maximize the detection accuracy using minimum number of features.

In majority of existing models that combine *BU* and *TD* processes, the *BU* and *TD* process are treated separately and at the end combined statically. For the feature weighting based models (e.g., [20, 22, 23]), the *TD* factors are used to tune the *BU* feature weights. These factors are not considered directly to decide whether such factors are important for a task or not. Similarly, in models where *TD* features are combined directly with *BU* features (e.g., [3, 28, 106]), the combination is static and again does not consider the importance and interaction between the two sets of features. Hence, the lack of interaction between the *TD* and *BU* processes typically results in feature redundancy, leading to poor accuracy and lower efficiency.

In FS-TDSM, we exploit the interaction between the features of both processes through feature selection modeling. Target object detection is chosen as an example to evaluate the performance of the model. A set of

BU and *TD* features are chosen therefore that make sense in the context of target object detection. The proposed model works in three steps: feature extraction, feature selection and saliency generation using the selected features. These steps are discussed below in detail.

5.2.1 Feature extraction

The mechanism used for feature extraction and integration is based on the *Itti* model [1]. However, instead of using three basic features, other features are included as shown in Fig. 5.1. The reason for using more features than Itti features is to have a richer collection of bottom-up features to perform a selection from, as a small number of features might not capture the interaction between the features. Furthermore, an investigation on whether (and how) the bottom-up features being used in this model generate better saliency maps than using the *Itti* model features is carried out.

There are two types of features being used in the proposed model, bottom-up features and top-down target specific features. The bottom-up features denoted as X_i for $i = 1, 2, \dots, P$, where P is the number of bottom-up features responsible for attending to salient regions of an image. A total number of 13 features are used as bottom-up features: red/green contrast, blue/yellow contrast, intensity, orientation measured at 0, 45, 90 and 135 degrees, green colour, blue colour, two principal component analysis (*PCA*) based features and two symmetry features.

Previously it has been shown that *PCA* features are very effective for detecting salient regions [76,94,120]. The process used in [120] is followed to extract *PCA* features from an image. Please refer to section 3.3.1.2 for details regarding the extraction of *PCA* features from a colour image. Empirically, the two principal components with the highest eigenvalues (denoted as d_1 and d_2) are selected.

The symmetric features are extracted using a local radial symmetry transform [146]. The symmetry is calculated over a circular window with

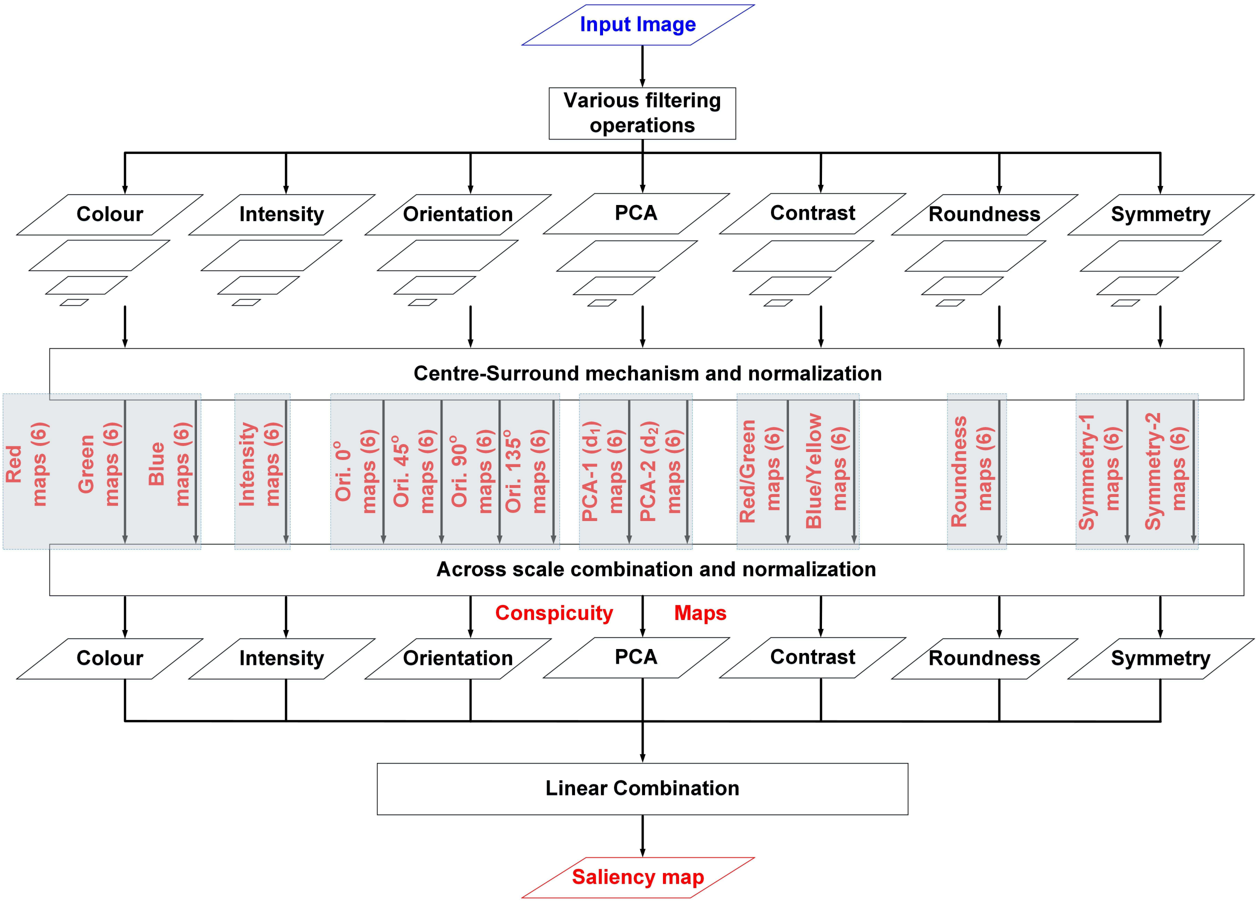


Figure 5.1: The proposed feature extraction and integration model for saliency map generation.

two radius values of 5 and 10 pixels resulting in extracting two symmetric features. The symmetric features serves two purposes. First, humans tend to fixate at objects in natural images that have certain geometrical symmetry as it has been hypothesized theoretically in [4, 85]. Furthermore, this feature in particular also acts as a target specific feature because the object under study in this work (i.e., the cricket ball) exhibits some level of symmetry around its seam (refer to sample images of the cricket balls in Fig. 5.7).

The second set of features are the top-down features indicated by Y_i, where $i = 1, 2, \dots, Q$, and Q is the number of top-down features. These features are chosen for a specific task. The target object to be detected is either characterized partially or fully by such top-down features.

As the target object to be detected is the red cricket ball, two target specific features are used, redness and roundness. Although these two features might not be sufficient to completely describe the target, the objective is to demonstrate the top-down process by a selection process between some bottom-up and top-down features. Note that how these two features describe the target object (i.e., the red cricket ball) compared to the bottom-up features as the red cricket ball is characterized by its colour (i.e., red) and shape (i.e., round).

In order to measure the roundness of objects in an image, the images are initially segmented using a contour based hierarchical segmentation technique [147]. Initially this technique extracts fine contours using a high performance contour detector that combines local and global image information. This is followed by a method to transform these contours into a hierarchy of homogeneous regions while maintaining the contour structure.

An upper size threshold and a lower size threshold are imposed on the thresholded image to eliminate those regions that might be generated due to noise, over segmentation or background uniformity. For each filtered

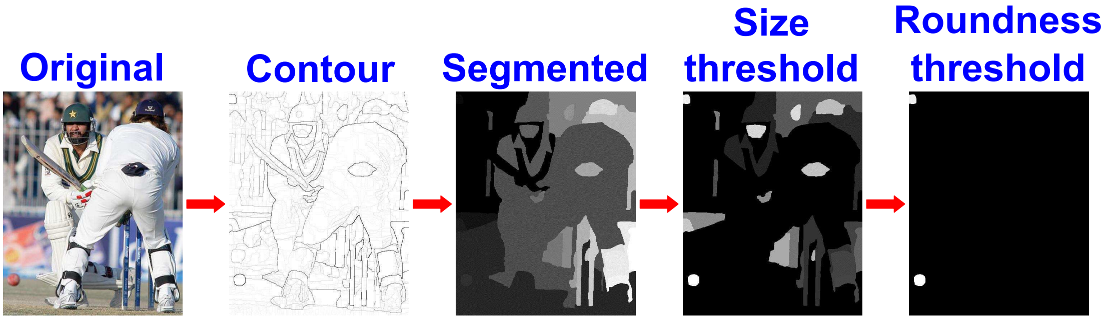


Figure 5.2: The steps for roundness feature extraction from an input image.

segmented region, the roundness is measured as follows:

$$\text{Roundness} = \frac{4\pi A}{P^2} \tag{5.1}$$

where A and P are the area and perimeter of the segmented region respectively. The segment is a perfect round object if the ratio given by Eq. (5.1) equals to one. Finally, after evaluating the roundness of the segmented region, these regions are further filtered using a threshold value for roundness. Any segmented region having a roundness value less than the threshold is removed. Figure 5.2 demonstrates the above-mentioned steps to generate the roundness feature for a sample image.

Once the features are extracted, the corresponding feature and conspicuity maps are produced as described in the *Itti* model.

5.2.2 Binary particle swarm optimization for feature selection

The second phase of the proposed model performs the feature selection over the set of top-down and bottom-up features. In the previous chapter, we have seen how particle swarm optimization (PSO) was able to successfully perform feature weighting to model contextual based top-down saliency. We again use PSO for the optimization process but now for feature selection. However, in this work, we use a different variant of PSO, referred to as binary PSO (BPSO) to perform feature selection [148].

Particle swarm optimization (PSO) was originally introduced for continuous search spaces, but Kennedy and Eberhart proposed another version of PSO that deals with discrete valued problems [149]. In the literature on feature selection, BPSO has been considered one of the successful techniques in performing features selection in a high dimensional feature space [148,150].

BPSO considers the position vector to have binary valued elements (i.e., either zero or one). In addition, the velocity given by Eq. (4.2) is normalized to a value between zero and one using a sigmoid function. Finally the position update equation given by Eq. (4.3) is replaced by the following:

$$\mathbf{x}_{kl}^{t} = \begin{cases} 1, & \text{if } S(\mathbf{v}_{kl}^{t}) > Z \\ 0, & \text{otherwise} \end{cases} \tag{5.2}$$

where S is a sigmoid function applied to the velocity and Z is some threshold value between zero and one. Note that the position vector of a particle represents the selected features while the velocity of a particle is one of the algorithm's characteristics that simulates the movement of a particle in the search space.

5.2.2.1 Feature selection

Binary PSO based feature selection considers the vector of positions to be a D-bit string where each bit controls whether a feature is included in the selection or not. When the values is one, it means that the feature is selected whereas zero means the feature is not selected. Many effective BPSO based feature selection algorithms have previously been proposed [148,150]. However, due to the small dimensionality of the search space in the this problem (15 features in total), a simple BPSO setup is sufficient for the task. Since the proposed model is extensible, increasing the number of features in the future might require a more effective BPSO algorithm.

5.2.2.2 Objective function

Figure 5.3 shows an example of generating a saliency map from the selected features and demonstrates how the fitness value is computed for a single image. The fitness function represents a saliency accuracy measure for evaluating the quality of the saliency map. As in chapter 4, we use the F-measure score for target detection evaluation. In the proposed model, the performance of a particle in BPSO (i.e., selected feature) is evaluated by calculating the F-measure of the saliency map generated from such particles. The pre-computed features for the corresponding image is loaded for each training image from the training dataset. Furthermore, only the selected features encoded by the D-bit string for the particle is used for the computation of the final saliency map. Note that the pre-computation of features for the training images is performed to make the objective function computation more efficient.

Once the saliency map is generated, it is segmented using an adaptive threshold value based on Otsu's method [151]. We use the Otsu's method instead of the adaptive mean-shift segmentation method used previously when computing the F-measure score (see page number 32 of chapter 2), for the following two reasons. First, Otsu's method is faster than the mean-shift segmentation as the latter performs an extra step of predicting a segmented region from the saliency map whereas no such step is required in Otsu's method. Secondly, as we will show in the results section that when the adaptive mean-shift segmentation is used, the selected features are different from those selected when Otsu's method is applied. When using the selected features from the mean-shift segmentation method on the test images, the detection accuracy was lower than that achieved by Otsu's method based features.

Once the saliency map is segmented, it is compared with the groundtruth to evaluate the precision and recall values. Finally, the F-measure between 0 and 1 is computed using Eq. (2.2) where the value of β^2 is set to 0.3 [74]. The F-measure calculation is repeated for all the images

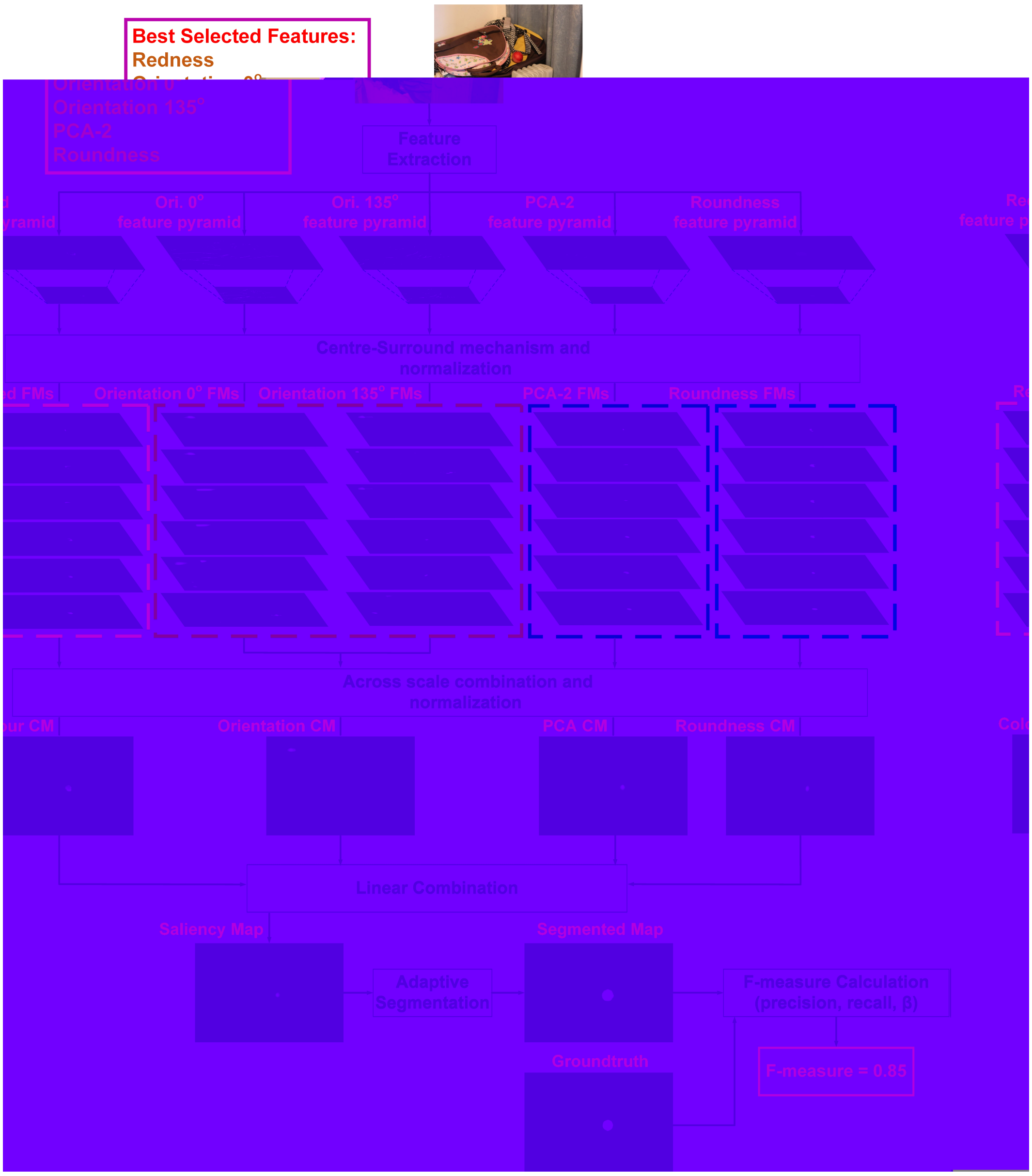


Figure 5.3: Fitness evaluation based on the selected features. The example demonstrates how the features, feature maps, conspicuity maps and the final saliency map are generated from the selected features. The example input image is from dataset five for which the set of selected features are redness, roundness, orientation 0°, orientation 135° and *PCA*-2.

and an average value is taken as the fitness value of the particle. Hence, the fitness function is the average F-measure over all the training images. Hence, the optimization problem is to maximize the F-measure value over all R examples in a training set.

This is different from the approach followed in the previous chapter when learning the optimized set of weights (i.e., the contextual based weight learning). In this model, the F-measure score is computed over all the images in the training set whereas in the contextual weight learning, the optimization of feature weights was performed on each image separately. Because we needed to perform a contextual association between the learned weights and the contextual descriptors of individual images, each image was considered for separate optimization during the training phase. In this model, because we want to have a single set of selected features for the entire dataset, we computed the average score over all the images in the training set. As a result, we expect that the time requires to evaluate the fitness of the particle in this setup is greater than that when performing individual image optimization.

5.2.2.3 Testing phase

In the testing phase, once the features are selected, they are used on the test images to produce the saliency maps in the same manner as in the training images. The F-measure is averaged over the testing images for a particular dataset to evaluate the models performance.

5.3 Simulation setup

5.3.1 Dataset

The proposed model is tested on five datasets with varying complexity and background content. The images in these datasets are a combination of self-created images and images collected from the internet. Each dataset is split into two subsets, 50% for training and 50% for testing. These datasets contain the target object (which is the red cricket ball) in different sizes, numbers, background and illumination conditions as described in Table 5.1.

In all five datasets, we only consider one type of target (i.e., the red cricket ball). Since we need to have certain explicit characteristic target features as top-down features, we confined our study to one simple object (i.e., the red cricket ball) which is well described by two features; the redness and roundness. More complex objects (for instance those considered in the seven datasets investigated in chapter 4) would require additional specific high-level features, however, this is not the objective of this chapter. Since the overall model is generic and independent of the object itself, we believe that it can be extended to any target object provided that a set of target specific top-down features are included in the optimization process.

To justify the creation of these additional datasets for our experiments, we investigated some of the existing standard datasets used in computer vision. In object recognition datasets (e.g., PASCAL VOC [52]) or salient object detection datasets (e.g., ASD [74]), the variation in the location of the target object and the contextual complexity is not considered. For instance, in salient detection datasets, mostly the objects are located in the centre of the image, whereas in the recognition datasets, the contextual and background content is rather simple as argued by the authors in [144]. Both these factors are considered in these five datasets, and hence they are suitable for analyzing the visual search process from both top-down and bottom-up perspectives.

Table 5.1: Details about the datasets which are used in the experiments.

Dataset	Instances	Size	Source	Description
Dataset one	50	400×300	Self created	Uniform outdoor illumination and background. Target is mostly non-salient with constant size. It has distracting objects.
Dataset two	28	400×300	Self created	Similar to dataset one but for indoor.
Dataset three	50	vary in size	From the internet	Outdoor cricket scenarios mostly with green grass. Target is mostly non-salient with variable sizes. Other objects present include players, bats, field and crowd.
Dataset four	50	vary in size	Self created and from the internet	Different scenarios with varying background. The target is purely salient and the size varies slightly. It has a small number of non-salient distracting regions.
Dataset five	42	vary in size	Self created and from the internet	Varying indoor illumination with cluttered background. The target is non-salient and varies in size. It contains complex distracting objects

5.3.2 Roundness parameters

As discussed is section 5.2.1, in order to extract the roundness feature from an image, various threshold values need to be set. The optimum values for these parameters are listed in Table 5.2. For both size parameters (i.e., upper bound (SSUB) and lower bound (SSLB)), the values are selected by computing the average ratio between the size of the target object and the size of the image in the respective datasets. For the roundness sensitivity parameter (RS), we have chosen a high sensitivity value so that the segmented region is only considered as a potential target cricket ball only when it has high roundness geometry. We have tested various values between 0.7 and one, and empirically came with the values listed in Table 5.2 that would maximize the detection accuracy.

5.3.3 PSO parameters

Table 5.3 summarizes the BPSO parameters used in the simulation. In [139], the authors used a population size of 100 particles to achieve good results in feature selection for classification with a number of features varying from 13 to 617 features. To have a better exploration of the search space with 15 features, we have increased the population size to 500. In the proposed model, it is more important to find a close to optimum solution with certainty than to get quickly to a local optimum solution.

Another parameter which is set for PSO is the neighborhood fraction (NF). The neighborhood fraction represents the fraction of the total population that participate in the update process for each particle. A small fraction of 0.1 is used to avoid stranding at local minima [142].

Each complete experiment was repeated 20 times with a random equal split of the training and testing sets. In each experiment, the process of feature selection using BPSO was repeated 30 times (i.e., the number of runs). Hence a total of 600 (20×30) experiments were performed on each dataset. The optimization at each run was executed for the complete number of al-

Table 5.2: Roundness feature extraction parameters. These parameters are learned for each dataset. The parameters are roundness sensitivity (RS), size sensitivity upper bound (SSUB) and size sensitivity lower bound (SSLB).

Dataset	**RS**	**SSUB**	**SSLB**
Dataset one	0.85	0.12	0.0050
Dataset two	0.80	0.10	0.0050
Dataset three	0.80	0.08	0.0005
Dataset four	0.80	0.30	0.0100
Dataset five	0.80	0.15	0.0005

Table 5.3: BPSO simulation parameters for feature selection.

Simulation Parameter	**Value**
Population size	500
Maximum number of iterations	50
Inertia weight	0.768
Acceleration constants	$c_1 = c_2 = 1.49$
Neighborhood fraction	0.1
Number of runs	30
Number of dataset partitions	20
Objective	Feature selection
Fitness function	Average *F*-measure
Stopping criteria	Fitness value of 1

lowed iterations (i.e., 50 iterations). For these experiments, the MATLAB optimization toolbox was used to implement PSO.

At each run, the best solution was evolved in the form of selected features. The overall best features having the best fitness value and the minimum number of features from all the runs are selected as the ultimate solution for the problem. The selected features can be pure top-down, pure bottom-up or any combination of both categories of features. The basic aim of BPSO is to remove the redundant features from the selection and exploit the interaction between various types of features to yield an optimum set of features having the best objective value.

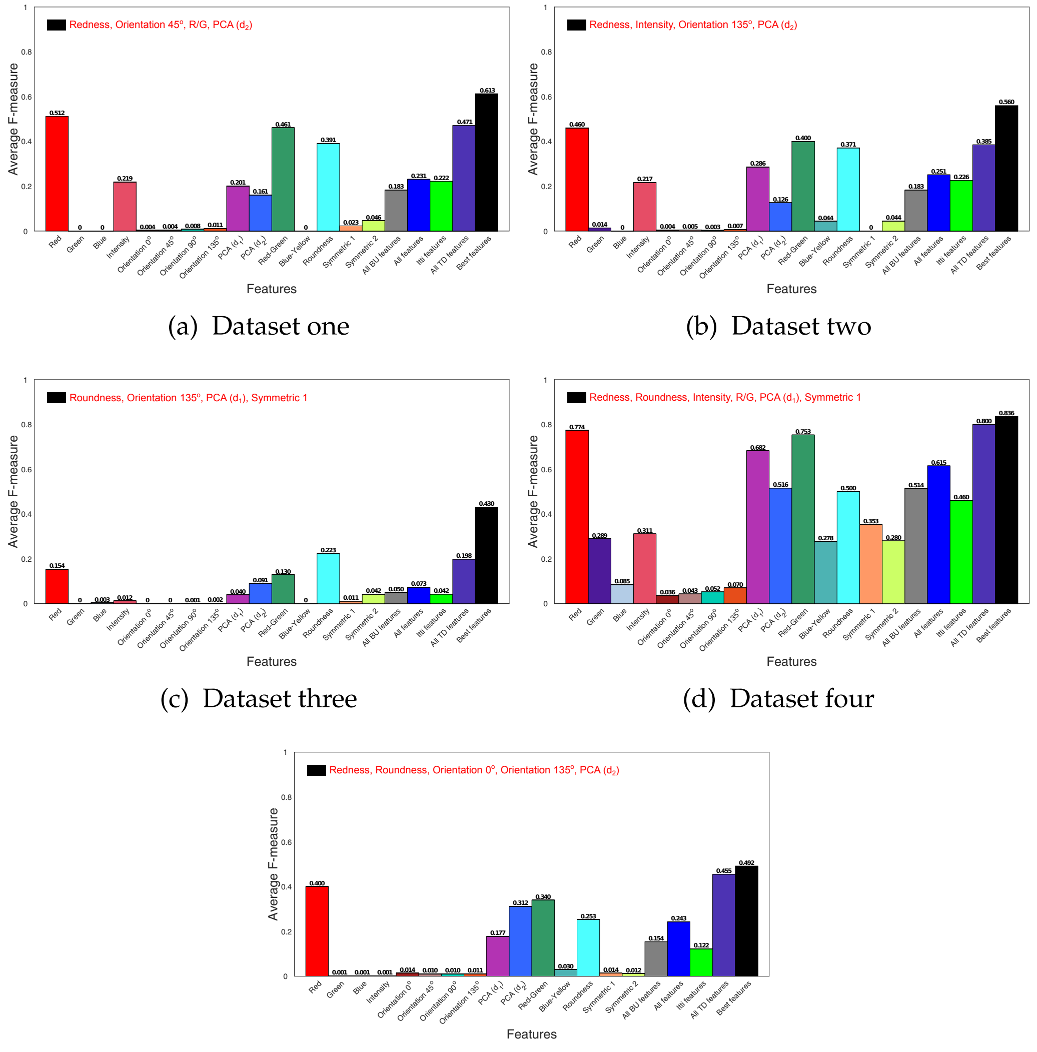
In some datasets, the selected features slightly varied from experiment to experiment. In such situation, the best-selected features from each experiment were ranked according to the number of occurrences. When a tie occurs, the feature with the highest reported fitness value was chosen.

5.4 Target detection results and discussion

The proposed model was tested on the five datasets and a group of features with highest *F*-measure fitness value are selected. The performance of the selected features was evaluated through their accuracy in detecting the target object. Figure 5.4 shows the best-selected features in each data set and the corresponding average *F*-measure fitness value reported on the training set.

5.4.1 *F*-measure fitness function results

Figure 5.4 shows the average *F*-measure score of various features when used to generate the saliency map on the test images. Higher *F*-measure scores indicate better detection of the target object. The figure shows the scores for all 15 features when used individually. In addition, the performance of the following feature combination is presented: all BU features,



(a) Dataset one

(b) Dataset two

(c) Dataset three

(d) Dataset four

(e) Dataset five

Figure 5.4: Feature profile of various feature combinations. The best selected features in each dataset are shown at the top left of each sub-figure. The fitness value of the best selected features are compared with that of individual features, when all the *BU* features are combined, when all the features (*TD* and *BU*) are combined and with *Itti* model features.

all features, *Itti* features and the selected features (referred in the figure as best features) obtained through the feature selection process. In addition, the figure also mentions the selected features in each dataset.

This result enables us to see the effect of combining *BU* and *TD* features in an ″optimal″ way and compare the performance with that of either pure *TD*, pure *BU* or a naive combination of both. In all datasets, it is evident from the figure that features selected by the proposed model exhibit highest *F*-measure score than all other feature combinations.

Amongst the selected features, there is at least one *TD* feature and more than one *BU* features that contribute in the detection of the target in each dataset. For instance, either redness or roundness is selected in the first three datasets, whereas both the *TD* features are selected in the last two datasets. Typically, in more complex images (e.g., in dataset five), *TD* features become more effective than BU features in detecting the target object.

To understand the level of interaction between *TD* and *BU* features, in Fig. 5.4 the *F*-measure score of the individual features from the set of best selected features are compared with that when they are combined. For the first and second datasets, the *F*-measure score achieved when using the selected features are 0.613 and 0.560 respectively. When only the redness *TD* feature is considered, the score are 0.512 and 0.460. respectively. Hence, an improvement of approximately 0.1 is achieved when certain *BU* features are added to the *TD* redness feature. In all other datasets, we observe a similar pattern and conclude that the performance is always improved when certain *BU* features are combined with the *TD* features. This shows the importance of *BU* features in detecting the target object.

Note that in the first two datasets, the *TD* roundness feature is not amongst the selected features. When this feature is considered solely, it achieves a good detection performance compared to the other features. Despite this, it has not been selected. This could be due to the fact that many distracting objects in these two datasets exhibit round geometrical structure and are falsely detected as target objects by this roundness feature. This is also evident when comparing with the ′All TD features′ (i.e., redness and roundness features). Hence, adding roundness to the redness feature degrades the performance slightly as many distracting objects in these two datasets are round in shape. Furthermore, when only the *BU* features are used (see the 'All *BU* features' in Fig. 5.4 (a and b)), the performance is degraded considerably, which suggests that pure *BU* is not a good option for target object detection in this case. Similarly, when all features are used, a slight improvement is achieved over the *BU* features, but the performance achieved is far from that of the optimized features.

In the third dataset, although similar results are achieved, there are a number of interesting points to be highlighted. First, the overall fitness (i.e., *F*-measure score) values achieved by the best features is lower than those achieved on the first two datasets. We expect this to happen as the background in this dataset is more complex and the size of the target object varies considerably. Secondly, the dominant *TD* feature here is roundness rather than redness as in the previous two datasets. The obtained fitness value when only the roundness feature is used is 0.223. This fitness is almost doubled when the best selected features are used (*F*-measure score of 0.430). This happens when certain *BU* features (orientation 135^o, *PCA*-1 and Symmetric-1) are combined with the roundness feature. This particular example shows how *BU* features can play a pivotal role in detecting the target object when appropriately combined with certain *TD* features.

As the complexity of the images through target location, the target view and the background complexity increases, the requirement to have more *TD* features increases. This is evident for dataset five in which both *TD* features are selected (see Fig. 5.4(f)). The significance of the *BU* features, in this case, is less than that in the first three datasets. Only an improvement of 0.092 is achieved when the *TD* roundness and other *BU* features are added to the redness feature. This result suggests that when the search for the target object becomes complex, *TD* factors become more effective in performing guided search than the *BU* ones.

Dataset four contains salient target objects with mostly simple background and no other distracting object. We expect that *BU* achieve good performance as these features are effective in detecting salient objects. As it is evident in Fig. 5.4(d) for some of the *BU* features (e.g., *PCA*-1, *PCA*-2, R/G). Even some of the *BU* features that have poor performance in other datasets exhibit better performance in this dataset (e.g., symmetry and green features). Although dataset four is less challenging than other datasets, both *TD* features are selected. This shows that even if the visual search for the target is simple, some influence of the *TD* targeted features saliency is normally useful.

Note that the performance of the best selected features outperform that of 'All TD features' in all datasets. This observation suggests that pure TD influence/features might not be the optimum solution when it comes to target object detection and some level of BU influence/feature might be needed to maximize the detection accuracy of the target object.

Finally, Fig. 5.5 examines the variation in the average fitness value over all the experiments for both training and testing datasets. The figure shows the average *F*-measure score along with the minimum and maximum scores across the datasets.

For all the datasets, the fitness variation and the reported average fitness values are consistent across the training and testing sets. The figure further shows that there is a small variation in the fitness values in dataset one, two and four for the different experiments. This is because both datasets one and two exhibit high visual contextual similarity. For dataset four, although the background of the images are typically different, the target always remains salient. If the target object is always salient, the selected features will be able to detect the salient object in most of the instances of the dataset because the target is usually located at the centre of the image. By contrast, considerable variation can be seen in dataset five which reflects the variability of the image contents in this dataset.

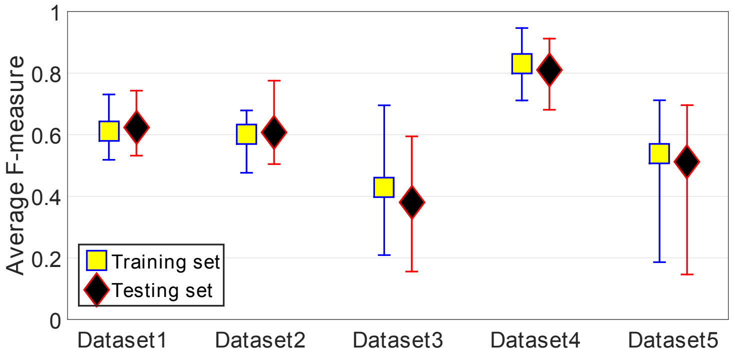


Figure 5.5: Average fitness value over all the training and testing images. The error bars span the range of minimum to maximum *F*-measure across the dataset. The maximum and minimum achieved fitness values is measured over all the optimization experiments with a different sampling of training and testing images.

5.4.2 Precision-recall accuracy results

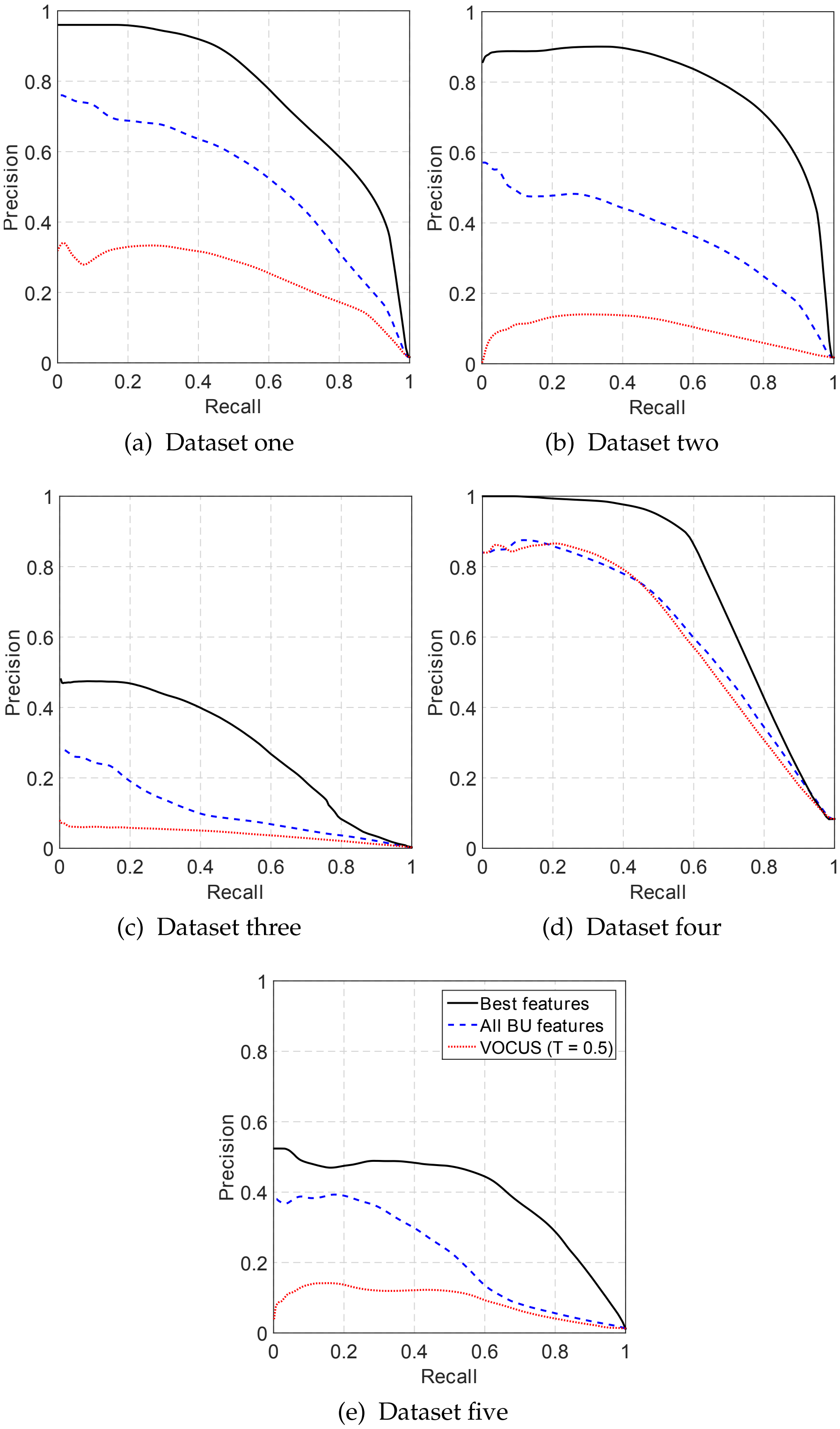
The precision-recall curve obtained with the selected features from the proposed model is compared with that for all features and with the popular top-down VOCUS model. When implementing the VOCUS model, instead of using only the original three *BU Itti* features, the *BU* features of the proposed model were used to ensure a fair comparison. Furthermore, the learned weights were averaged over all training examples and not over limited examples as discussed in [106]. This is because the average of *F*-measure values in the proposed model is taken over all the training examples. Finally, the combination factor T in Eq. (2.8) is set to 0.5 to ensure the equal involvement of both *TD* and *BU* maps when combined linearly.

Figure 5.6 shows the average precision-recall curves obtained over all the test images. In each dataset, we can see that the best selected features from the proposed model *FS-TDSM* outperforms the *BU* features and the VOCUS model. The VOCUS model performed poorly in all datasets other than the relatively easy dataset four. As mentioned earlier, this could be due to the method of modeling *TD* through feature weighting of the *BU* features without including any target specific features. Similarly, when all the features are used, we hypothesize that the performance degrades due to the presence of redundant features or because of the negative interaction between certain features. Such feature redundancy is reduced by the proposed model by exploiting the interaction between features. The precision-recall performance achieved by the proposed model in the last two datasets is lower than the performance in the first two datasets due to contextual variation of the images in those datasets.

5.4.3 Qualitative target detection results

To supplement the quantitative results, Fig. 5.7 shows examples from the datasets, the corresponding saliency maps and the adaptively segmented maps using the approach of [151]. Two examples are selected from each dataset. For the first dataset (see the first two rows), the target object is distracted by other objects positioned at random positions. The target in both images is slightly occluded by the non-uniform surface. Distracting objects in these two images (and normally in all the images belonging to this dataset) do not exhibit high visual attribute similarity to the target object. The baseline *Itti* model was able to fixate on the target object but with many false positive (FP) regions, as shown by the corresponding saliency and segmentation maps. A similar visual outcome can be seen from VOCUS model, which produces fewer FP regions but assigns the target only low saliency. The proposed model was able to detect the target in both images with high precision and accuracy.

The next two images from the second dataset contain cluttered distracting circular structured objects and can, therefore, be expected to be more challenging. Furthermore, the target in the first of these two images is non-salient whereas, in the second one, it is slightly salient. From the information about the selected features in this dataset, the roundness feature is not selected because most of the distracting objects are round. The proposed model was able to detect the target easily in the second of these two



(a) Dataset one

(b) Dataset two

(c) Dataset three

(d) Dataset four

(e) Dataset five

Figure 5.6: The precision-recall curve of the proposed model *FS-TDSM*, the top-down VOCUS model and the all *BU* feature combination for all the datasets. The proposed model outperforms the other two approaches in all the datasets.

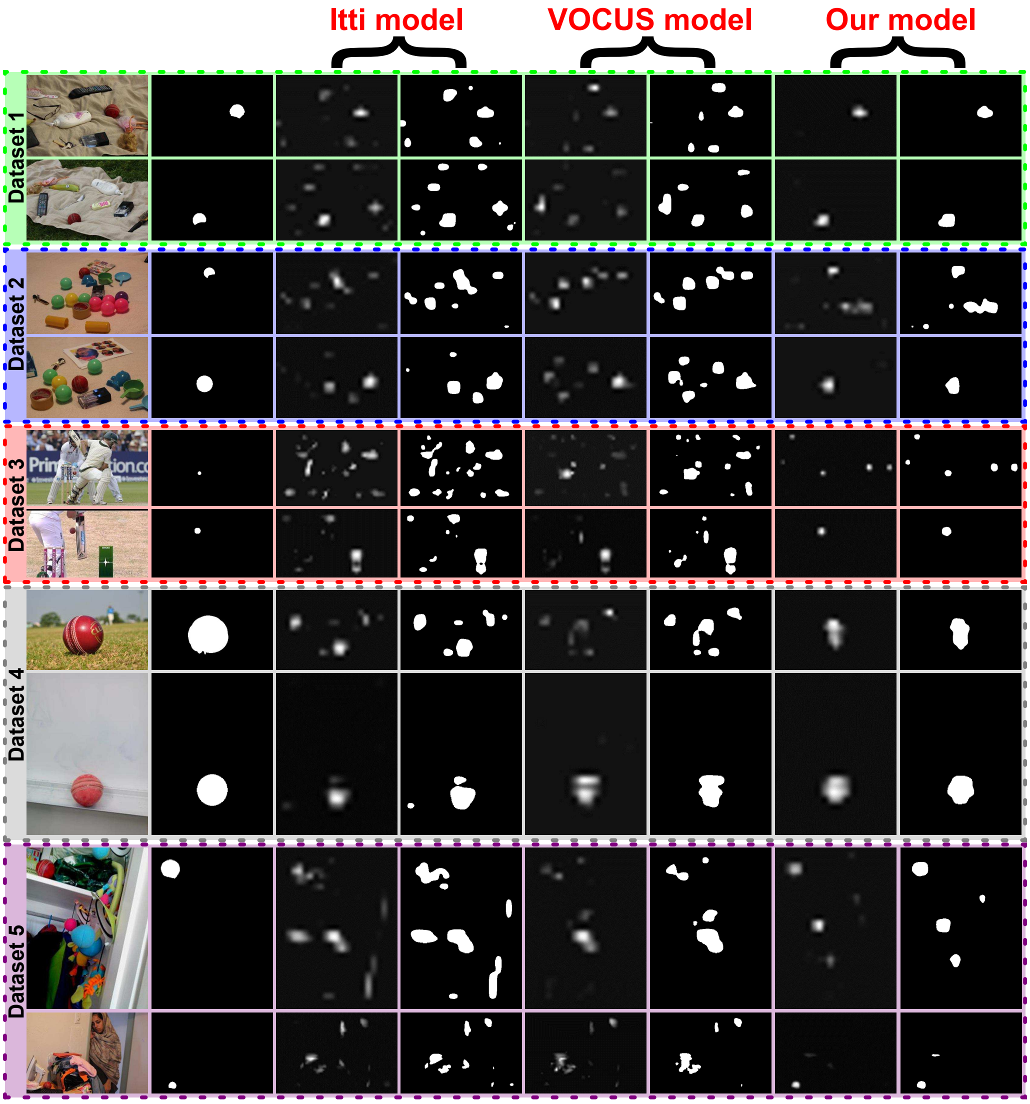


Figure 5.7: Qualitative comparison of the saliency maps for target object detection. Two images are selected from each dataset (i.e., the first and second row images are from dataset one, the third and fourth row images are from dataset two and so forth). The columns from left to right are the original image, groundtruth, *Itti* model saliency maps, Segmented maps of *Itti* model, VOCUS model saliency maps, Segmented maps of the VOCUS model, the proposed *FS-TDSM* model (best features) saliency maps and the segmented maps.

images, however in the first image, apart from detecting the target, those objects having some level of redness are also been selected (see last two images of the third row in Fig. 5.7). This particular example shows that in any attention system, if the objective is to detect a target object, then the type of *TD* target specific features that are required for detection depends on the image content (e.g., nature of distractors, context, etc.).

In the images drawn from dataset three the target size is relatively small compared to other objects. In both images (see row five and six), by comparing the segmented image produced by both *Itti* and VOCUS models with the respective groundtruths, we can see completely scattered regions of interest that contain the target but also a high number of false positives. The proposed model, on the other hand, was able to detect the target more accurately. The saliency map generated by *FS-TDSM* for the first image of this dataset contains five regions of interest including the target. The false alarm regions do not have any visual appearance reflecting the redness. This is true as redness is not amongst the selected features (see Fig. 5.4(c)). The selection of these false regions are due to some of the selected *BU* regions and the *TD* roundness feature.

All the techniques performed well in the two images from dataset four. This is due to the simplicity of the dataset and because the target object is salient. Since all the models (including *FS-TDSM*) use the *Itti* model structure which is based on the centre-surround difference of Gaussian procedure at various scales, the saliency maps are blurry and provide blob type fixation regions rather than object based regions.

For the above reason, the generated saliency maps in this dataset exhibit low recall values due to large false negative regions (e.g., saliency map produced by *FS-TDSM* for the image in row seven). Large false negative are resulted due to large spatial regions the target object occupy (e.g, the image in row seven).

Finally, the last two images, drawn from dataset five are the most challenging ones in Fig. 5.7. For instance, both *Itti* and VOCUS models were not able to locate the target due to clutterness in the background in the last image. The proposed model effectively located the target using the appropriate selected features.

5.5 Fixation and visual search results

In this section, the capability of the proposed model in performing a visual search for the target object is discussed.

5.5.1 Quantitative visual search results

From the previous results, it is evident that the proposed model is able to select an optimized set of features containing both *TD* and *BU* features. Furthermore, the selected features indicate that a positive interaction of various levels exists between both *BU* and *TD* factors depending on the nature and the contents of images. Although these results are sufficient to make a conclusion about the usefulness of both *TD* and *BU* factors in a high-level target object detection task, it remains unclear whether the detected region of interest containing the target is due to *TD* or *BU* features.

The optimized selected features for a particular dataset only provides the information that certain features give an overall good detection result over all the examples in that dataset by minimizing the false negative regions and maximizing the true positive ones. It does not say which of the selected features contribute the most in localizing the target region.

In some occasions, the saliency map produced by the proposed model highlights the target region but with a low associated saliency value (e.g., the saliency map generated by *FS-TDSM* for the fifth row image in Fig. 5.7). This result suggests that if the attention and fixation are applied to this image based on the saliency map, then the target is not detected in the first fixation. However, the target could be well detected in the subsequent fixations. Hence, one of the results to be discussed in this section is the number of fixations and saccadic movements required to get to the target (i.e., to perform an effective guided search) by the proposed model.

Two well known biologically inspired operations called winner-take-all (WTA) and the inhibition-of-return (IOR) are used to control the gaze shift from one attended location to another on the basis of a saliency map [18] (refer to section 2.3.1.2.1 for details about these two mechanisms). By applying these two operations on the final saliency map, we get the winner location that has the maximum activation in a particular sample region within the saliency map. We use the same WTA and IOR approach proposed in the *Itti* model [1]. Once a region is selected through the WTA network and IOR, a region growing based segmentation technique is applied to estimate and segment the selected region. This process is repeated for different regions of the image according to the saliency map. Each winner represents a fixation region whereas the estimated path between any two consecutive fixations represents the saccade or the scan path.

Table 5.4 compares the visual search accuracy for the proposed model with that of the VOCUS model. The number of fixation represents the number of regions that have been attended when the statistics are calculated. For instance, with a single fixation, we are concerned only with the first attended region generated by the WTA and IOR operations on the saliency map. Typically, as the number of fixations increases, the chance of detecting the target also increases. The detection accuracy is measured by the hit rate, which is defined as a region of fixation that overlaps with 50% or more of the groundtruth map.

As shown in Table 5.4, when a single fixation is considered, the VOCUS model operated at $T = 0.5$ achieves a 32% hit rate on dataset one. When the number of fixations is extended to three, a big jump to 88% is achieved followed by a 100% success rate when the attention is observed over five fixations. These numbers show the effectiveness of VOCUS model in finding target objects in a small number of fixations. However, a more impressive performance is achieved by the proposed model *FS-TDSM*. In a single fixation, the proposed model was able to achieve 96% hit rate. This confirms the effectiveness and efficiency (i.e., fewer fixations) of the model when performing a visual search on this dataset. A similar performance is observed on dataset two for the proposed model. In both these datasets, we can observe that for the proposed model, the first fixation typically produces high success rate and reaches the maximum rate of 100% within three fixations. Such behavior reflects the high quality of the initial saliency map generated by *FS-TDSM*.

For dataset three, the first fixation hit rate is low compared to the first two sets when *FS-TDSM* is used. When the number of fixations is increased (i.e., three, five and seven), the success rate is almost constant. This could be due to two reasons. First, since the majority of success is attained with a single fixation, mostly the other WTA/IOR regions are false alarm regions. Secondly, the maps generated by the proposed model for this dataset are concise. Few regions of interest exist that mostly belong to the target region and being already captured by the first fixation.

Hence, by increasing the number of fixations, false positive regions or regions that have already been explored by previous fixations are accessed. Revisiting a target region by increasing the fixations still counts as a single hit. This could be the reason why there is no considerable improvement beyond the first fixation.

In dataset five, the proposed model clearly outperforms the VOCUS model with a maximum hit rate of 96% from the fifth fixation and beyond. Even with the clutterness and background complexity of this dataset, the model was able to search and locate the target quickly. In all the discussed datasets, as expected, we can see that *FS-TDSM* achieves high success rates within the first seven fixations and outperforms the VOCUS model.

In dataset four, we can observe a different behavior. The hit rate achieved by *FS-TDSM* is almost the same throughout for all the fixations. Although this is a simple dataset in which the target object is salient and clear, the maximum achieved hit rate is 80% which is low if we consider the simplicity of the dataset. A hit rate of 100% is achieved in some of the datasets more complex than this one.

Table 5.4: Fixation accuracy performance of the visual search for the target object. For each number of fixations, the fixation performance of the VOCUS model and the proposed model *FS-TDSM* are presented indicated by ′A′ and ′B′ respectively. Each value represents the percentage of images in a particular dataset for which the visual search produced a hit in detecting the target object within a specific number of fixations. The two numbers between brackets in bold show the contribution percentage of bottom-up and top-down features respectively when the target is found successfully. This is measured by counting the number of hits that are due to any of the features belonging to bottom-up or top-down process during the visual search process.

	Datasets				
Fixations	Dataset one	Dataset two	Dataset three	Dataset four	Dataset five
One	A: 32%	A: 0%	A: 8%	A: 52%	A: 0%
	B: 96%	B: 72%	B: 58%	B: 76%	B: 53%
Three	A: 88%	A: 36%	A: 40%	A: 56%	A: 48%
	B: 100% **(40)(60)**	B: 100% **(37)(63)**	B: 72%	B: 80% **(46)(54)**	B: 91%
Five	A: 100%	A: 57%	A: 52%	A: 60%	A: 62%
	B: 100%	B: 100%	B: 76% **(25)(75)**	B: 80%	B: 96% **(19)(81)**
Seven	A: 100%	A: 79%	A: 68%	A: 60%	A: 67%
	B: 100%	B: 100%	B: 76%	B: 80%	B: 96%

Such performance on this particular dataset could be due to the following reasons. The way the saliency is computed (based on the multiple scale DoG centre-surround mechanism), tends to make the maps blurry while fixating on the centre of the region of interest. Hence, such techniques partially highlight the object of interest with high intensity at the centre and lower values in the surrounding region. This is also reflected in Fig. 5.6(d) where it is clearly evident that the precision is very high for low threshold values and drops significantly at an approximately 0.6 recall value. This typical behavior occurs when some parts of the object of interest are detected with high saliency and other parts with low saliency. Furthermore, when the size of the target object is relatively large with respect to image dimension, only a small part of it is detected with high saliency. Other parts of the object are detected as false negatives.

Only a small part of the target is extracted when the segmentation step is performed. Since the hit is counted only when 50% of the target is detected, those small extracted regions of the target are ignored. Similarly, the performance of the VOCUS model is low because of the previously mentioned reason (since it is also based on *Itti* model) as well as due to the many false positive regions in the saliency map.

The target objects in the fourth dataset are salient but with varying sizes. For instance, in the example shown in Fig. 5.7 (see row seven and eight), the ratio of the target size and image size is different. By inspecting the saliency maps generated by *FS-TDSM* across this entire dataset, it is found that the target was detected accurately with high precision but low recall when the target size is large and again detected accurately with high precision and high recall when the target is small. We can verify this by comparing the saliency maps generated by *FS-TDSM* for the two images belonging to dataset four in Fig. 5.7.

Finally, for the maximum achieved hit rate in each dataset, Table 5.4 also shows the percentage of the hits being due to either *BU* winners/features or *TD* winners/features. This information reveals which process (*BU* or *TD*) has the main contribution in finding the target.

For instance, in dataset one, 40% of the success can be attributed to the selected *BU* features (i.e., either orientation^0, R/G or *PCA*-2) whereas 60% is contributed by the *TD* redness feature.

By observing the results in the other datasets, we can see that the highest *BU* contribution of 46% is in the saliency dataset. This comes as no surprise because *BU* factors always play an important role in saliency detection (for target or non-target objects). The *BU* contribution is typically smaller when the images are complex. This is when *TD* factors become more effective in a more guided search for a target rather than simply using *BU* features to attend to locations which are deemed to be interesting. However, despite the low contribution of the *BU* factors in such images, there is always a positive role of such factors in detecting the target (e.g., 25% and 19% hit rate by *BU* features in dataset three and five respectively).

5.5.2 Qualitative visual search results

As an example of the level of contribution made by individual *BU* or *TD* features in searching for the target, Fig. 5.8 supplemented by Table 5.5 show the fixation at various locations of the images and the winner feature at each fixation region. A total of seven fixations and six saccadic movements are shown such that the saccadic sequence is displayed next to the respective saccade path. In addition, when the fixated region produces a hit, a black contour is drawn through a region growing segmentation procedure otherwise it is shown in yellow. The winning feature that produces a hit is highlighted in bold in the table. Five different images from the datasets are chosen to investigate various scenarios. The proposed model is compared with *Itti* model, all *BU* features and VOCUS model.

In the first image, all four approaches successfully found the target but after a different number of fixations. Both the *Itti* and the VOCUS models located the target in the third fixation and with a similar saccadic profile. When all the *BU* features are used (i.e., no redness or roundness features),

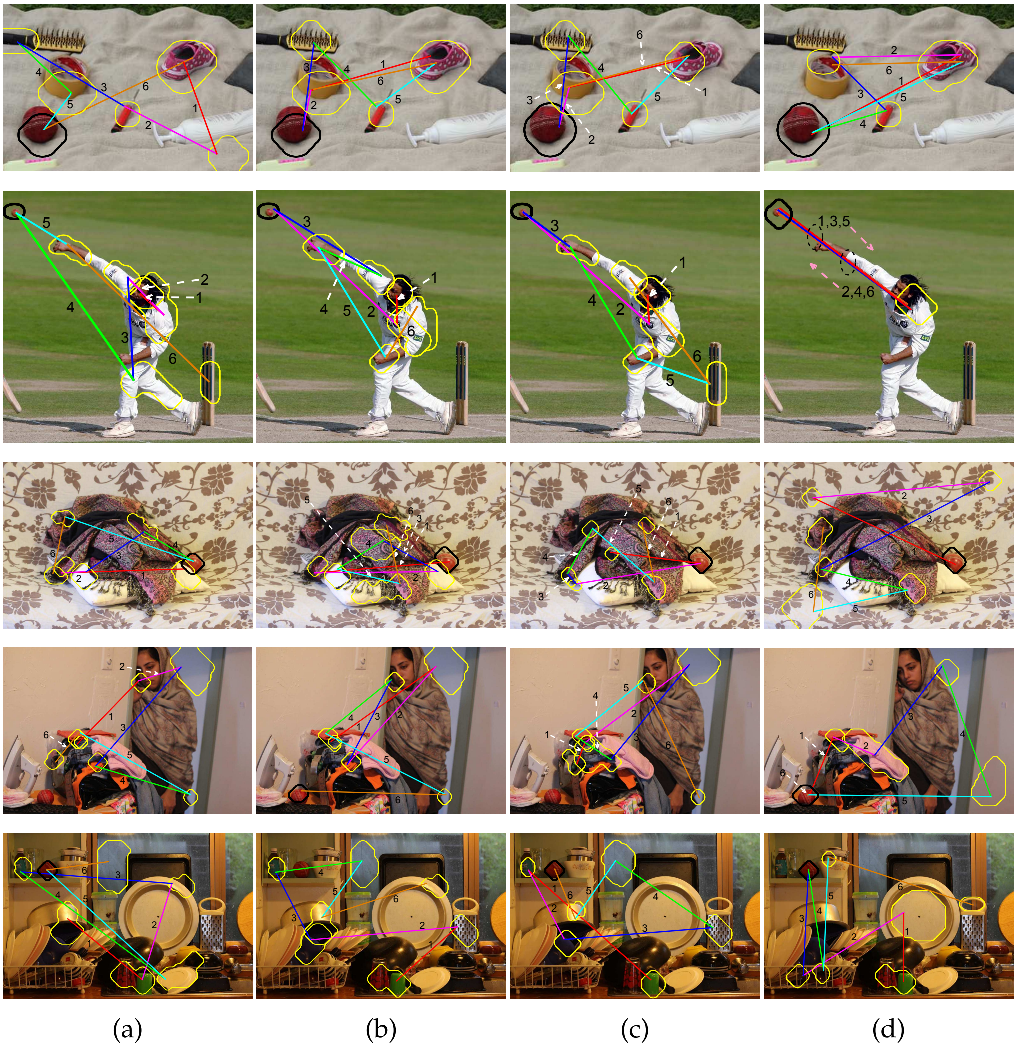


Figure 5.8: Saccadic movement examples during a visual search for the target object. The images exhibit different level of complexity through background clutterness, distracting objects, target location and size. A black contour shows a hit when the target object is located. The qualitative performance is compared amongst (a) *Itti* model (b) All *BU* features (c) VOCUS model (d) The proposed *FS-TDSM* (best features).

	Approach			
Saccade	**Itti model**	**All BU features**	**VOCUS model**	**FS-TDSM**
		Image 1		
1	B/Y	B/Y	Blue	**(Redness)**
2	Intensity	PCA-2	PCA-2	PCA-2
3	B/Y	**(Intensity)**	**(R/G)**	Redness
4	Intensity	Symmetry-1	Symmetry-1	Redness
5	R/G	PCA-1	R/G	**(Redness)**
6	**Orientation (0^{o})**	B/Y	Blue	PCA-2
7	B/Y	PCA-2	PCA-2	Redness
		Image 2		
1	R/G	PCA-2	R/G	**(Roundness)**
2	Intensity	Symmetry-1	Symmetry-1	Orientation 135^{o}
3	Intensity	**(PCA-2)**	**(PCA-2)**	**(Roundness)**
4	Intensity	Intensity	R/G	Roundness
5	**(R/G)**	PCA-2	PCA-2	**(Orientation 135^{o})**
6	R/G	PCA-2	Blue	Roundness
7	B/Y	Intensity	Intensity	PCA-1
		Image 3		
1	Intensity	**(PCA-2)**	Green	**(PCA-2)**
2	R/G	R/G	**(PCA-2)**	Roundness
3	Intensity	Intensity	Green	Roundness
4	R/G	Symmetry-2	Blue	Redness
5	**(R/G)**	Intensity	Green	PCA-2
6	Intensity	B/Y	Blue	Orientation 0^{o}
7	Intensity	B/Y	Blue	Redness
		Image 4		
1	B/Y	B/Y	Green	**(Roundness)**
2	Intensity	B/Y	Blue	PCA-2
3	B/Y	B/Y	Blue	PCA-2
4	B/Y	Intensity	Blue	PCA-2
5	B/Y	PCA-1	PCA-1	Orientation 135^{o}
6	R/G	B/Y	Symmetry-1	**(Roundness)**
7	Intensity	**(PCA-2)**	Blue	**(PCA-2)**
		Image 5		
1	Intensity	PCA-2	Green	PCA-2
2	R/G	Symmetry-1	Green	Roundness
3	Intensity	B/Y	Blue	**(Roundness)**
4	B/Y	B/Y	Symmetry-1	Redness
5	Intensity	PCA-1	Blue	Orientation 0^{o}
6	**(R/G)**	Intensity	PCA-1	PCA-2
7	B/Y	Intensity	**(R/G)**	Roundness

Table 5.5: Individual winning features corresponding to the example images in Fig. 5.8 for the proposed model and other approaches. The features in bold represent those winner features that achieve a target detection when performing a visual search.

the search process took longer to locate the target (i.e., sixth fixation) which suggests that some of the *BU* features used might have a negative impact on the search process. The proposed model, on the other hand, was able to locate the target twice during the search process; on the first fixation and revisited the target on the fifth fixation. On both occasions, the winner feature is the *TD* redness. Furthermore, this example by no mean suggests that the false attended regions before fixating on the target are due to any of the *BU* features. For instance, the first two false fixation regions by *FS-TDSM* are due to *TD* redness and *BU PCA*-2 features respectively.

Going through the second image in Fig. 5.8, when the proposed model was used, only two regions are attended. The target object is detected alternatively between these two regions as shown in the figure (see the fourth image in the second row of Fig. 5.8). Interestingly, the winning feature changes as shown in Table 5.5. In the first two hit occasions, the *TD* roundness feature is responsible for the detection whereas the orientation 135^o and *PCA*-1 *BU* features corresponds to the third and fourth positive fixations on the target.

In the previous two images, the *TD* features for the proposed model dominated the successful fixation on the target object. The third image, on the other hand, reveals an important point regarding the *BU* features and their possible contribution to the visual search. Regardless of the number of fixations required to locate the target, in the all *BU* features, the VOCUS model and *FS-TDSM*, the winning feature responsible for detecting the target is *PCA*-2 (see Table 5.5 and the corresponding fixation in Fig. 5.8). In addition, we can clearly see that the first fixation that located the target in the proposed model is due to a *BU* feature and not a *TD* feature.

In the fourth image, which has a more complex background, we can observe that neither VOCUS nor the *Itti* model was able to fixate on the target during the first seven fixations. The all *BU* features detected the target in the last fixation through the *PCA*-2 feature. By contrast, *FS-TDSM* was able to effectively detect the target on three occasions, fixation one and six due to roundness feature and the last fixation due to *PCA*-2.

In the final image which is probably the most complex example in this figure, all the approaches struggled to find the target within the first three fixations. The contrast R/G feature produces a hit in both the *Itti* and the VOCUS models but in the last two fixations respectively. Although the R/G feature in both these models has different set of features to interact with, in both cases such interaction resulted in the detection of the target. Furthermore, a better performance is achieved by the proposed model in locating the target in the fourth fixation. Since R/G feature was not amongst the selected features for this image which belongs to the fifth dataset, no target fixation based on this feature is evident when *FS-TDSM* is applied.

Finally, from the fixation and visual search experiments, we clearly see that the proposed model produces a high success rate in all the datasets when performing a saccadic guided search for the target, outperforming both the VOCUS and the *Itti* models. Furthermore, on most occasions, the proposed model was able to locate the target within the first few fixations. This shows the effectiveness and efficiency of the model in locating the target as quickly as possible. Furthermore, although *TD* features play a more significant role in the search for the target than *BU* features, there is clear evidence both qualitatively and quantitatively that *BU* features are more effective than *TD* features in some scenarios particularly in the visual search process.

5.6 Chapter summary

In this chapter, an attentional model is proposed that combines bottom-up and top-down saliency processes to improve the target object detection accuracy over using either of the individual processes. The proposed model called Feature Selection based Top-down Saliency Model (*FS-TDSM*) formulates the combination process as a feature selection problem. Several top-down target specific and bottom-up attentional features are included in the pool of features to perform the selection upon.

The model is tested on five datasets containing cricket balls as target objects with varying image content complexity. The detection accuracy is evaluated through the average *F*-measure score over all the images in the test set. In addition, the visual search accuracy and efficiency is determined through the number of fixations required to get to the target object. The proposed combination model is compared to one of the state-of-the-arts visual attention model referred to as VOCUS [20] that combines top-down and bottom-up processes statically.

With a set of two target specific top-down features and 13 bottom-up features, our proposed model outperformed the VOCUS model both in detecting the target object as well as achieving higher efficiency (fewer number of fixations to get to the target object) while performing a visual search. The scope of the model is valid only when target specific features are combined with low level bottom-up features through feature selection process. If only bottom-up or top-down features are considered during the selection process, then it does not address the visual attention combination problem.

The proposed model provides high interpretability of the selected features. From the obtained selected features in each dataset, it has been observed that the features from both processes are amongst the selected features. The selected features in all datasets do not have features that are purely belonging to either bottom-up or top-down. Hence we conclude that, both processes contribute actively while performing a visual search for the target object. The level of contribution varies depending on the contents of the image. With images with moderate complexity (low cluttered images) with few or no distracting objects, the contribution of bottom-up features increases. This contribution decreases in more complex background images but do not diminish. In such cases, top-down features are more effective in detecting the target objects.

Our model is only tested on a single object (i.e., the cricket ball), and could be potentially generalized to any object provided that a set of distinct top-down and bottom-up features are available. Furthermore, our feature selection approach for combining top-down and bottom-up saliencies can be used to construct a more complex and efficient visual attention systems for any target object detection. As a future work, to see the effectiveness of our model, it would be interesting to apply it to various types of objects.

Chapter 6

Dynamic Feature Map Integration for Visual Attention

6.1 Chapter introduction and motivations

In the previous chapter, a feature selection approach was proposed to combine top-down (TD) and bottom-up (BU) saliency features. The proposed model provided set of features to include in the combination process. When a test image is processed, the previously selected features are used to generate the saliency map. In fact, the same selected features are used for all the images in a dataset. With this approach, there is no guarantee that the selected features represent the optimized solution for each individual image. To overcome this problem, a revised model is needed that can decide which features to combine and when to combine them on an individual image basis. In this way, a dynamic approach of combining top-down and bottom-up saliency features can be achieved.

In the work described in this chapter, we seek to find an approach to estimating a quality measure value of a saliency or feature map by extracting some useful information from the map. That is, we attempt to find whether the nature of the map is itself indicative of its quality. Furthermore, we show that if the quality of a feature map is predicted accurately, a dynamic feature map integration can be achieved. A dynamic feature map integration adds more efficiency to the visual attention system in addition to improving the detection accuracy.

As described in chapter 2, a saliency or feature map is a 2D topological map on which high activation points correspond to novel regions that are deemed pertinent to the machine's intended task. One might intuitively expect that the form of the map should be meaningful. For example, a very diffuse map would likely be poor for guiding future choices, as it would not do a good job at identifying areas of particular interest. Similarly, maps that exhibit a multitude of isolated salient regions are not particularly informative.

If we could characterize the quality of a map by examination of the map itself, then we have the capacity to dynamically improve the saliency map when combining BU and TD features (later we will refer to them as feature maps). For instance, features of the image could be incorporated (or removed) until the saliency map assumes a suitable form (i.e., a good quality saliency map). Using this approach we would no longer be tied to using the potentially unnecessarily expensive set of predefined features (as it was the case when formulating the combination process through feature selection), but could instead develop a feature set in real time.

Determining the quality of a saliency map has been addressed in the previous literature on active vision [12, 32, 109, 152]. However, our model learns a quality metric of a saliency map unlike other techniques that use deterministic methods to measure the quality of a saliency map. In addition, the proposed model can be used to determine the quality of any saliency map, whether it belongs to BU or TD saliency map. Note that in all our previous proposed models presented in chapters 3-5, once a saliency map is generated, there is no mechanism that would examine how good the generated map is for target object detection.

Typically the saliency map produced by a proposed set of features is compared to some groundtruth map and characterized by a metric such as

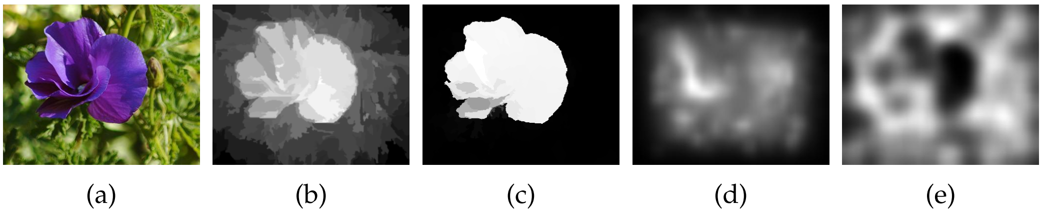


Figure 6.1: Visual difference between object detection saliency maps and fixation maps. Fixation maps tend to be blurry and dispersed as in (d) [91] and (e) [134]. On the other hand, object detection saliency maps are compact and sharp as in (b) [92] and (c) [133].

the area under the receiver operating curve (AUC-ROC). A saliency map that exhibits a high agreement with the groundtruth map is considered a good map. Such a comparison is only applicable when the groundtruth map is available. In the absence of a groundtruth map, estimating a quality metric value of a saliency map becomes a challenging task. Therefore, our goal would be to estimate an AUC-ROC score of a given saliency map of a novel image.

There are many active vision tasks that might potentially benefit from the quality estimation approach, but unfortunately, that diversity produces a wide range of differing practical details. However, irrespective of the task or the application, the visual appearance of a saliency map is highly dependent on the process itself for generating the saliency map (e.g., see the visual difference of the four saliency maps in Fig. 6.1). Hence, rather than directly tackling the full range of potential problems, in this chapter, we therefore confine ourselves to a single case study to establish the merits of the approach.

We have selected the pedestrian detection problem treated in [3] because of the well-characterized set of features used to derive the saliency. Hence, the main objective of this chapter is to show that a dynamic feature combination process for detecting the target object is possible on run time if we are able to estimate the quality of a saliency map.

6.1.1 Terminology

In this chapter, various overlapping terminologies are used to describe different aspects of the proposed work. To avoid confusion, in this section we explicitly define these terms. The scope of these terms are only within this chapter. When an image is presented to the proposed model, various features are extracted. We refer to *features* as those values that are extracted from images such as Gabor filter based orientation features, pedestrian detection features and contextual features. This term is not frequently used in this chapter.

More importantly, when these features go through some post processing, 2D topological maps are generated that we refer to as *feature maps*. Throughout this chapter, we will be using three types of feature maps based on the model proposed by Ehinger et al. [3]. It is possible that a single feature map is generated through various features (depending on the type of the feature map). When all the feature maps are combined, a *saliency map* is generated, which also represents a 2D topological map. Note that for instance, if two feature maps are combined, the resultant is still referred to as a feature map because it exhibits the same 2D topological structure as a single feature map or a saliency map.

The main objective of the model is to estimate a quality score of any map (either feature or saliency map). To achieve this, a set of different kind of features are extracted from a map. To distinguish these features from the features discussed earlier, we refer to the latter as *characteristic features*. Hence, *features* are extracted from the original image to help in constructing the *feature maps* whereas *characteristic features* are extracted from the *feature maps* or the *saliency maps* for estimating their quality.

Note that extracting the characteristic features, the process does not differentiate between a a feature or saliency map because both are 2D topological maps. Hence, unless mentioned explicitly, we use the term feature map throughout the chapter to indicate that the characteristic features are extracted from such maps and can be generalized to saliency maps as well.

Finally, we use the descriptive term *visual attributes* to explain the visual look of a feature or saliency map. In this work, we hypothesize that the visual attributes of a map can summarized by extracting the proposed *characteristic features* of these maps.

6.1.2 Chapter objectives and overview

This chapter examines two main aspects of the overall problem,

1. Do the visual attributes of a feature or saliency map provide any information about its goodness through some defined quality metric?

2. If the visual attributes of a feature or saliency map are informative of its quality, then how effectively can this knowledge be used for dynamic selection of such maps?

In order to achieve the above two objectives, we perform the following:

1. We posit a set of characteristic features that potentially encapsulate desirable visual attributes of a feature map. In subsequent testing, we establish a subset of these characteristic features that are most useful for our application.

2. From the characteristic features of a feature map, we learn a regression model in a supervised framework that estimates the quality of a novel feature map. The regression model is learned from a large set of training examples.

3. We conduct a study to demonstrate how the visual attributes of feature maps can be used in dynamic feature map selection. The regression model is incorporated into a feature map integration framework proposed by [3] in order to validate its effectiveness for dynamic feature map selection. In contrast to the approach in [3], which always

combines three constituent feature map subsets, we construct a system that progressively builds the saliency map and allows a shortcut as soon as the saliency map is of adequate quality.

Hence, the proposed work in this chapter establishes the following important contributions to the field of active vision:

1. Propose a set of effective characteristic features that are used to capture the visual attributes of a feature map.

2. Propose an approach that can estimate a quality score of a feature map using random forest regressor through characteristic features.

3. Analyze the learned regression model along with the characteristic features by investigating their importance and their relation with the estimated variable on the fixation dataset.

4. Use the proposed estimation approach with a feature map integration framework to validate its effectiveness in dynamic feature map selection.

6.2 Previous quality prediction techniques

Several techniques have previously been proposed to estimate the quality of a saliency map for either fixation or salient object detection. Probably the initial attempt in addressing the quality of a saliency map could be seen in the *Itti* model itself [1]. In this model, the normalization process prior to feature map integration acts as a way to suppress those feature maps that have homogeneous activation regions. At the same time, the normalization process promotes maps that contain small number of activation points.

In another set of techniques referred to as complementary saliency maps choose one or more feature map/s having the best performance.

Such selection varies from one image to another. For example, Gopalakrishnan et al. [152] proposed a saliency usefulness measure called the saliency index (SI) that is computed from two characteristic features, compactness, and connectedness. Inspired by this work, Cheng et al. [130] introduced global cues that utilize the compactness characteristic feature of the SI measure for feature selection through a soft image abstraction procedure. A more sophisticated compactness measure was proposed by Kim et al. [32] that combined various map information including boundary, location, brightness and background priors for selecting the best map.

Hu et al. [109] proposed a saliency measure called the Composite Saliency Indicator (CSI) that is used to assess whether an area within a generated feature map is likely to be a true candidate salient region. The indicator is a combination of two characteristic features extracted from a feature map, the spatial compactness, and the saliency density. Feature maps with a CSI value lower than a threshold are removed from the feature map integration process. Finally, each feature map is weighted by its CSI value and the feature maps are combined together to form the saliency map.

In all these techniques, the compactness is used as one of the prominent characteristic features for measuring the quality of a saliency map in salient object datasets. In these datasets [74, 82, 153], the salient objects are located at the center of the image and are highly focused. As a result, the generated saliency maps tend to be compact and localized [27]. However, not all saliency maps having high-level compactness correspond to good prediction capability. A map that has a false positive region with high compactness establishes a poor detection performance. In addition, since fixation maps are mostly spatially diverse, the compactness characteristic feature becomes less effective in describing the quality of these maps.

To illustrate this issue, we performed an experiment to show that compactness does not necessarily provide true information about the quality of a saliency map. As shown in Fig. 6.2, a saliency detection technique [33]

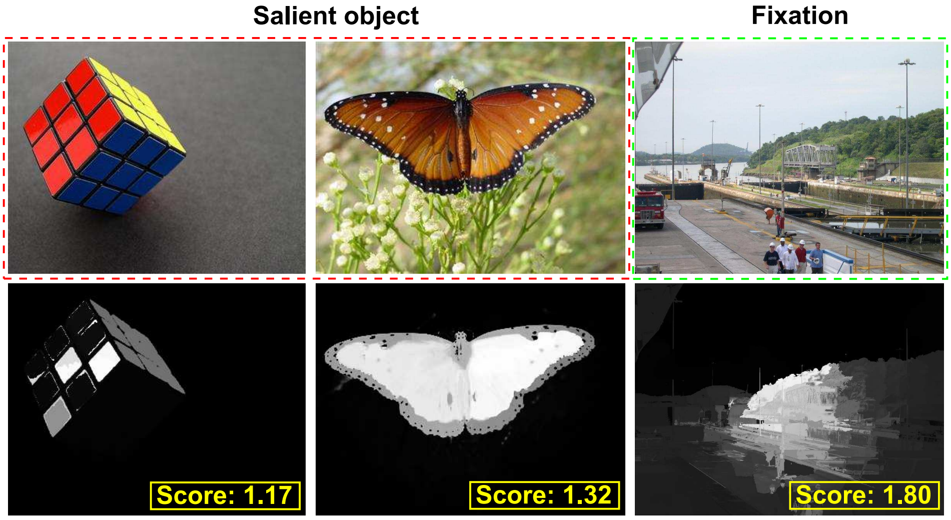


Figure 6.2: Compactness measure for the saliency and fixation maps. The example shows that compactness does not always correspond to good detection performance, particularly in fixation maps. The fixation map does not provide a good detection although it received the highest compactness score of 1.80 compared to the other two saliency detection maps.

based on measuring the compactness of a map was employed on two images from a saliency object dataset [74] and another from a fixation dataset [3]. As it is evident from the maps of the saliency object detection images, a higher compactness score corresponds to a better quality saliency map and in turn better detection performance (compare the butterfly map with that of a Rubik's cube). In the third image, although it is evident that the saliency map displays a poor detection performance (the groundtruth represents the people in the image), it receives the highest compactness score.

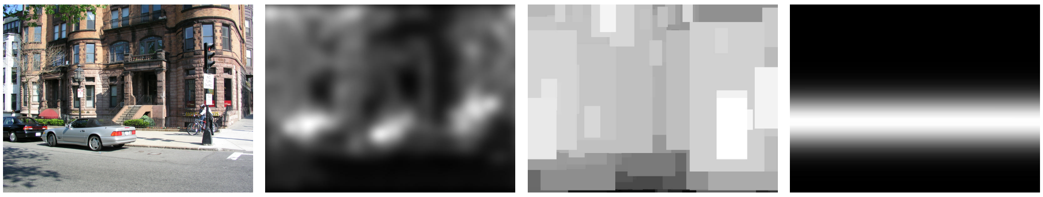
In another approach developed by Li and Itti [12], a machine learning technique was used to classify small patches extracted from a large satellite image as positive (indicating the presence of a target) or negative classes. After generating the saliency map of a patch, two sets of features called saliency and gist features were extracted. From these sets of features, a feature vector of 238 dimensions is constructed that captures the local and global visual attributes of a single feature map. The visual attributes are a combination of low-level information (intensity, orientation, and junction points) and statistical information (mean, standard deviation, the number of local maxima points and the Euclidean distance between them). A binary classifier was trained from these feature vectors from various training chips using support vector machines (SVM) for target presence or absence in each patch. The authors show promising performance in detecting different target objects accurately in various satellite images.

The Li and Itti method does not explicitly describe the quality of a saliency map but rather summarizes a saliency map using a feature descriptor which is further used for target detection through binary classification. Although the model can be applied to both saliency and fixation maps, it has two limitations. First, it is computationally expensive to construct the feature descriptor as it uses the attention model proposed by Itti et al. (the *Itti* model), which is slow due to multi-scale processing and multi-level centre-surround operation [1]. Secondly, the authors did not consider any geometrical or positional features in their descriptor and hypothesized that the descriptor is position, scale, and rotation invariant. This could be valid for a small region of an image (i.e., as a local descriptor), but can not be generalized globally.

6.3 Case study

Specifically, we investigate our proposed model under a feature map integration framework proposed by Ehinger et al. [3]. This framework belongs to a category of feature map integration technique called saliency aggregation. The initial version of this technique integrated several BU feature maps to improved salient object detection accuracy. Each BU feature map is different in the sense that it is generated through a different algorithm. Each feature map captures a different aspect of the salient region.

Ehinger et al. [3] on the other hand asserted that by combining differ-



(a) (b) (c) (d)

Figure 6.3: An example of visual appearance of the three source maps generated by Ehinger et al. [3]: (a) original image (b) bottom-up saliency map (c) target map and (d) contextual map.

ent types of feature maps (e.g., bottom-up, top-down, target and contextual maps), better accuracy in predicting human fixation can be achieved. Their assumption is based on the notion when human analyze a visual scene, they perform multiple incremental perceptual tasks, each having its own importance [154].

As a demonstration, three feature maps were used to predict where humans fixate when searching for people in outdoor images. Figure 6.3 shows a typical visual appearance of these three feature maps generated for an image [3]. The first source is a pure data-driven bottom-up feature map generated through a statistical saliency model proposed by Torralba et al. [119]. This map captures what is deemed unique or interesting in a scene.

The second feature map is a pure top-down map called the target feature map (TM). Since the objective of the guided search in this application is to look for pedestrians, a high-level state-of-the-art pedestrian detector proposed by Dalal and Triggs was used [62]. The produced map reflects the probability that a pedestrian is present within a particular search window. More information about the features, training setup, classification model and other tuning parameters for detecting pedestrians can be found in [3, 62].

The final feature map is the scene context (CM), which describes the overall holistic appearance of a scene. Based on the framework proposed by Oliva and Torralba [2] and Torralba et al. [119], an overall representation through global features that describes the spatial frequencies and orientations of images was developed for scene context. Influenced by top-down constraints, the generated map captures regions that have high likelihood of containing people (e.g., sidewalks or roads rather than sky or trees). The visual appearance of the map shows a horizontal strip that corresponds to a region where people can most often be found. After learning appropriate weights for each feature map, the weighted feature maps are summed to form the final saliency map.

Ehinger et al. hypothesized that the performance in predicting human fixation is maximized by integrating the three feature maps [3]. This is true when the performance is averaged over all the images in a dataset. However, we found that when inspecting individual images, combining all feature maps does not always provide the best solution. For instance, Fig. 6.4 shows that the AUC-ROC score and the hit ratio of TM alone, CM alone and TM combined with CM are higher than the combination of all maps (see section 6.4.1 for details of the AUC-ROC metric).

We hypothesize that if an appropriate combination of maps on an individual image basis is exploited, the overall prediction accuracy will be increased as irrelevant or misleading feature maps will be excluded from the combination process. Hence, in this chapter we explore how an estimated quality score of a feature map can be used to achieve more effective and dynamic combination of feature maps.

6.4 The proposed approach for feature map quality estimation

In this section, we investigate two key components of our proposed approach for estimating a quality score of a feature map. The first is the characteristic features that are extracted from a feature map and used to

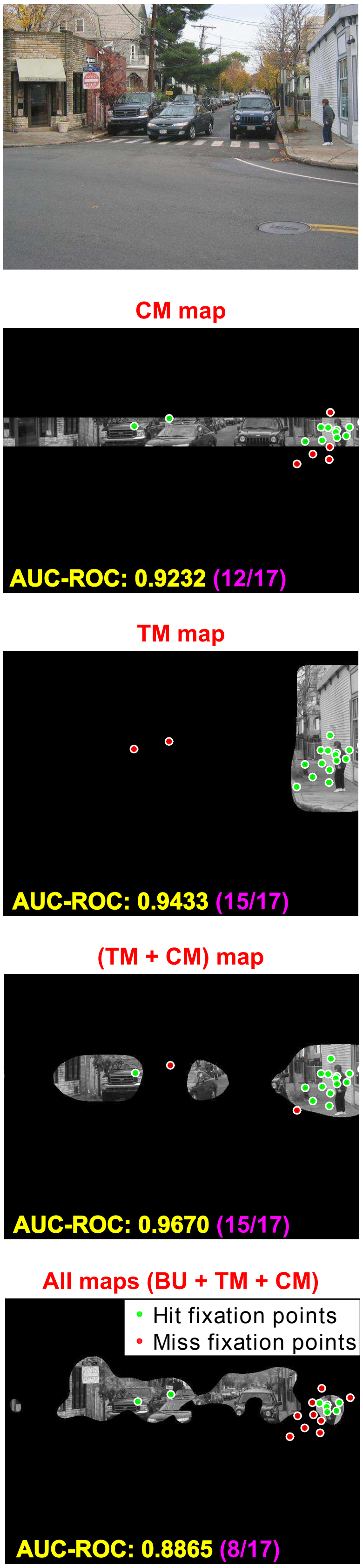


Figure 6.4: A visual comparison between all and partial feature map combination. The number between brackets shows the ratio between the human groundtruth fixation points (green points) that overlap with the predicted saliency region and the total fixation points.

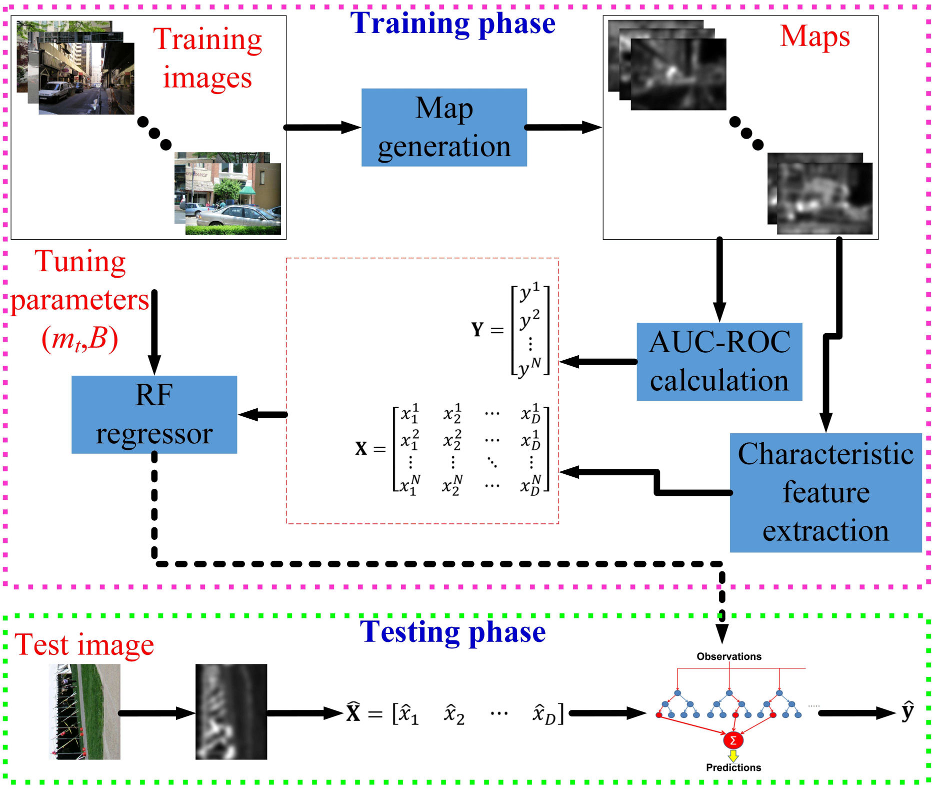

Training phase
Training images
Map generation
Maps
Tuning parameters (m_t,B)
RF regressor
$\mathbf{Y} = \begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^N \end{bmatrix}$
AUC-ROC calculation
$\mathbf{X} = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_D^1 \\ x_1^2 & x_2^2 & \cdots & x_D^1 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^N & x_2^N & \cdots & x_D^N \end{bmatrix}$
Characteristic feature extraction
Testing phase
Test image
$\hat{\mathbf{X}} = [\hat{x}_1 \quad \hat{x}_2 \quad \cdots \quad \hat{x}_D]$
Observations
Σ
Predictions
$\hat{\mathbf{y}}$

Figure 6.5: The training and testing phases of the proposed feature map quality estimation approach.

train a regressor. The second component is the regression model itself using a random forest regressor.

6.4.1 General structure of the proposed approach

The novelty of our proposed approach lies in presenting the problem as a regression problem where the outcome variable represents the AUC-ROC value of a feature map. As mentioned before, the AUC-ROC score represents the quality metric used to measure the goodness of a feature map.

Figure 6.5 illustrates the training and testing phase of our approach. In the training phase, N feature maps of some type are generated from N

training images. A set of D characteristic features are extracted from each feature map to form a characteristic feature vector $\mathbf{x}_i = [x_1^i, x_2^i, \ldots, x_D^i]$ where i is the training image index. The empirical AUC-ROC of a feature map is computed as a function of threshold θ and given by:

$$\begin{aligned} \mathrm{FAR}(\theta) &= \frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}}, \quad \mathrm{DR}(\theta) = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \\ y &= \mathrm{area}(\mathrm{FR}, \mathrm{DR}) \end{aligned} \tag{6.1}$$

where false positive (FP), true negative (TN), true positive (TP) and false negative (FN) are information extracted from the feature map by comparison with groundtruth. The false alarm rate (FAR) and detection rate (DR) are computed over several threshold values given by θ and area is a function to estimate the area under the parametric ROC curve as θ is varied.

Hence, given a training dataset $\mathcal{D} = (\mathbf{x}_i, y_i), i = 1, \ldots, N$, the regressor learns a model H such that $\hat{y} = H(\hat{\mathbf{x}})$ for a set of characteristic features $\hat{\mathbf{x}}$ from a novel feature map. In the next section, we explore the regression model itself in more detail.

6.4.2 Characteristic feature extraction

One of the key elements of the proposed approach is the characteristic features extraction process. The characteristic features should be simple, efficiently computed and should collectively capture the spatial distribution, geometrical structure and positional information of the feature map both in the local and global sense.

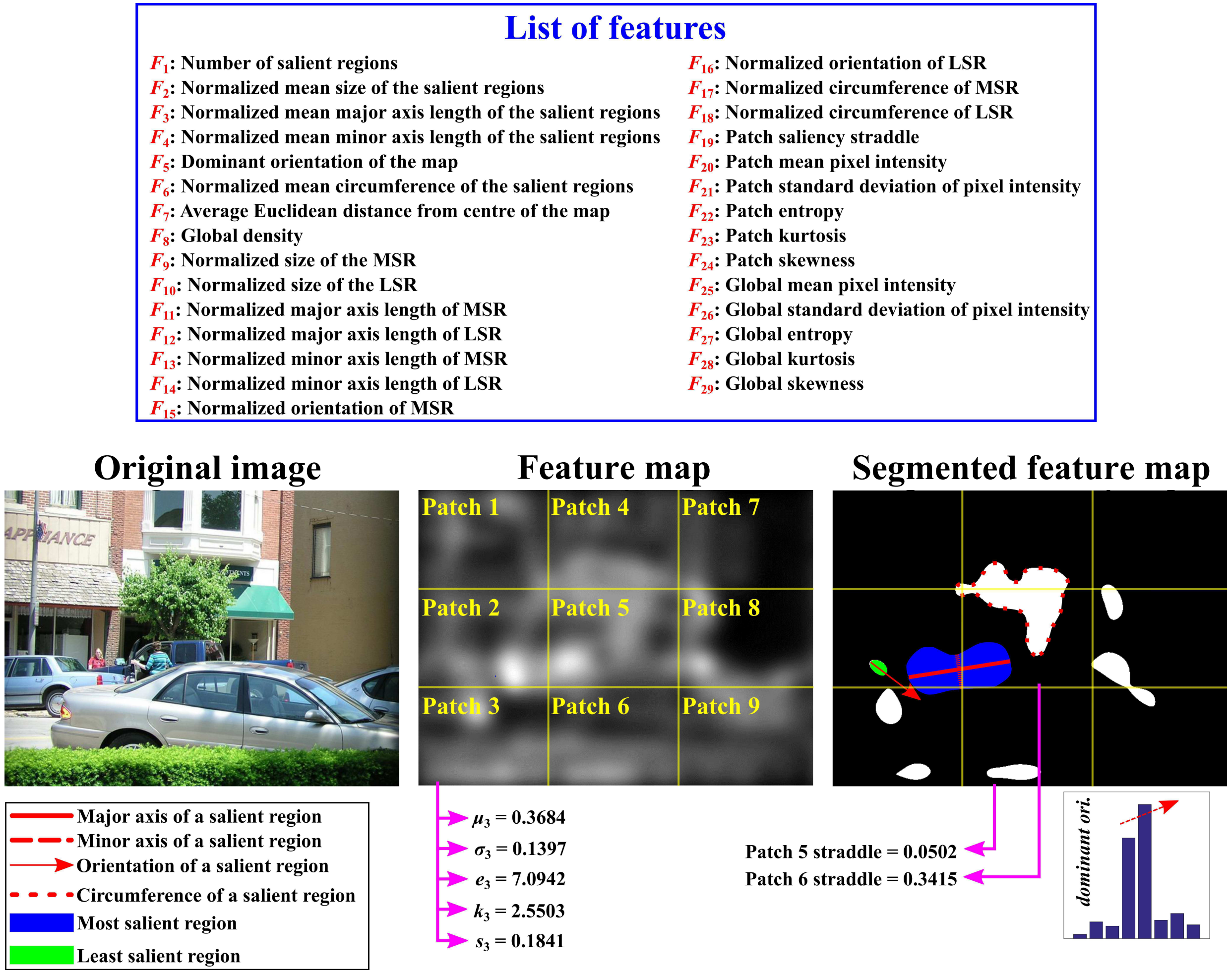
To acquire such data from a feature map, a total of 29 characteristic features $(F_1, F_2, \ldots, F_{29})$ are extracted and a characteristic feature vector of length $D = 23 + 6P$ is constructed $(f_1, f_2, \ldots, f_D)$ where P represents the number of local patches used. Out of 29 characteristic features, 23 characteristic features are extracted over the entire feature map. The other six characteristic features are extracted from P patches of the feature map, yielding $6 \times P$ characteristic features, one for each patch.

Patches represent rectangular non-overlapping regions in a feature map as shown in Fig. 6.6. Extracting characteristic features at the patch level provides local information in a map. Increasing the number of patches results in extracting finer local details from a feature map. However, increasing the number of patches reduces the efficiency of the characteristic feature extraction process. In all our simulations, various patch sizes were used, however, $P = 9$ is considered to balance between fine information extraction (and better representation) and computational speed.

As shown in Fig. 6.6, the characteristic features are extracted from two maps, the actual feature map and its segmented version. The feature map is segmented by applying a fixation threshold value K to retain the most salient regions. This threshold represents the fraction of the feature map area that needs to be retained (i.e., to be segmented). Once the K percent activation points from the entire feature map are retained, the saliency map in converted into a binary map using a binarization threshold value. This value is set to the smallest value of the retained activation points. Any activation point on the feature map less than this value is set to zero and one otherwise. Note that the value of the activation points in the feature map is normalized between zero and one. As in [3], we select $K = 10$ in all our experiments. The details about the characteristic features are given below. Each characteristic feature has its significance in determining the quality of a feature map.

1. **Geometric global statistics**: these characteristic features are used to describe a geometrical shape of the overall map from the segmented salient regions. A single mean value over all the salient regions for each characteristic feature belonging to this category is computed. These characteristic features are:

   - Normalized mean size of the salient regions (F_2). The size of a salient region is measured by the number of pixels within a



List of features

F_1: Number of salient regions
F_2: Normalized mean size of the salient regions
F_3: Normalized mean major axis length of the salient regions
F_4: Normalized mean minor axis length of the salient regions
F_5: Dominant orientation of the map
F_6: Normalized mean circumference of the salient regions
F_7: Average Euclidean distance from centre of the map
F_8: Global density
F_9: Normalized size of the MSR
F_{10}: Normalized size of the LSR
F_{11}: Normalized major axis length of MSR
F_{12}: Normalized major axis length of LSR
F_{13}: Normalized minor axis length of MSR
F_{14}: Normalized minor axis length of LSR
F_{15}: Normalized orientation of MSR
F_{16}: Normalized orientation of LSR
F_{17}: Normalized circumference of MSR
F_{18}: Normalized circumference of LSR
F_{19}: Patch saliency straddle
F_{20}: Patch mean pixel intensity
F_{21}: Patch standard deviation of pixel intensity
F_{22}: Patch entropy
F_{23}: Patch kurtosis
F_{24}: Patch skewness
F_{25}: Global mean pixel intensity
F_{26}: Global standard deviation of pixel intensity
F_{27}: Global entropy
F_{28}: Global kurtosis
F_{29}: Global skewness

Figure 6.6: Computation of some of the characteristic features extracted from a feature map. The geometrical and positional related characteristic features are extracted from the segmented feature map whereas pixel based characteristic features are extracted from the main feature map. The figure lists all the characteristic features that are used in the regression model. The values of the patch based characteristic features for some of the patches are also provided.

segmented salient region.

- Normalized mean major axis length of the salient regions (F_3). It is a scalar quantity that measures the length of the larger axis of an ellipse that encloses the salient region. The ellipse and the segmented salient region have the same normalized second central moment.
- Normalized mean minor axis length of the salient regions (F_4). This is same as F_3 but for the minor axis of the ellipse.
- Dominant orientation of the map (F_5). Orientation of a salient region is the angle measured between the major axis of the salient region and the horizontal axis. The dominant orientation represents the overall orientation of the map as shown in Fig. 6.6. It is calculated by constructing a weighted histogram of salient regions' orientations similar to that constructed in histograms of oriented gradients (HoG) descriptor [62]. The weights represent the normalized area of the salient regions.
- Normalized mean circumference of the salient regions (F_6).

All the size based characteristic features (i.e., F_2, F_3, F_4 and F_6) provides information about the average size of the target object to be searched for. For instance, in the dataset used in our experiments, the target object (i.e., pedestrians) are relatively small with respect to the size of the image. In addition, the major and minor axis of a pedestrian follow certain proportion. In a similar manner, the dominant orientation characteristic feature provides the angle at which the target object is usually found. In case of pedestrians, they are usually found in an upright position.

2. **Pixel level global statistics**: this set of characteristic features gathers pixel level intensity information of the whole map. The following characteristic features were considered:

- Global density (F_8). We use the same density extraction mechanism proposed by the authors in [109] which is given as:

$$F_8 = \frac{1}{|\mathcal{S}|}\sum_{z\in\mathcal{S}} \frac{\sum_{r\in\mathcal{S}_p} |G(z)| - |G(r)|}{|\mathcal{S}_p|} \tag{6.2}$$

  where $\mathcal{S}$ is the set of most salient points in a feature map, $\mathcal{S}_p$ is the set of all the neighboring salient points to p and G represents pixel intensity value. This measures the density of a feature map according to the saliency intensity within a pixel neighborhood.

- Mean saliency intensity (F_{25}) and standard deviation of the intensity values (F_{26}).

- Global entropy (F_{27}). It measures randomness of a feature map. High global entropy feature maps indicates high contrast in the map.

- Pixel intensity kurtosis (F_{28}) and skewness (F_{29}) are used as higher order moments to capture some information regarding the shape of the saliency distribution of a feature map.

These global pixel wise characteristic features are useful in describing the overall visual attributes of the feature map. These characteristic features do not describe the actual desired visual attributes of the target object. For instance, we would typically expect a good feature map to have high contrast with high entropy and to be more dense.

3. **Geometric MSR/LSR statistics** (F_{9-18}): this category of characteristic features is similar to the global version except that the features are not computed over all salient regions but rather on the most salient region (MSR) and least salient region (LSR) separately. The most salient region is defined as the segmented region having the high-

est mean intensity. For instance, the blue region in Fig. 6.6 has the highest mean pixel intensity whereas the green region is the LSR.

The reason for considering MSR and LSR is that if the target object represents the most salient region of the image, then these characteristic features would explicitly capture its saliency attributes. In the contrary, LSR characteristic features are useful only when the target object exhibits the least saliency in the entire image. So MSR and LSR characteristic features consider special saliency cases of the target object.

4. **Pixel level local statistics**: this characteristic feature set is computed as in the global version but on a smaller local region defined by a patch. Certain regions/patches in a feature map may contain more useful characteristic features than those extracted the whole feature map. In addition, saliency information may vary from patch to patch. For instance, if the object of interest is located in the center of the image, then the center patch would be more informative than the surrounding patches. Hence, each patch may have different visual attributes that could be useful in feature map representation.

   Apart from mean pixel intensity (F_{20}), standard deviation of the intensity (F_{21}), entropy (F_{22}), kurtosis (F_{23}) and skewness (F_{24}) calculated at the patch level, another characteristic feature is introduced that measures the straddle of all the salient regions over a patch (F_{19}). This characteristic feature describes the saliency attributes of a patch through salient regions in the segmented map. It is given as:

$$F_{19}^{j} = \frac{\mathrm{area}\left(\bigcup_{i=1}^{Z}(S_i \subset l_j)\right)}{\textit{Patch area}} \tag{6.3}$$

   where j is the index of a patch l and S_i represents a particular segmented salient region from a total of Z salient regions. Equation 6.3

represents the fraction of the patch that contain salient regions. The area of the salient regions that straddle into the patch is divided by the size of the patch.

Along with the straddle characteristic feature, Fig. 6.6 shows other sample characteristic feature values extracted from some of the patches. Note that only saliency straddle is computed on the patches of the segmented feature map (see patches five and six) whereas the rest are extracted from the patches of the original feature map (see the third patch). The reason for having patch wise characteristic features because it provides information about each region of the image. For instance, in many cases, the lower three patches of the images in this dataset are more likely to contain the street or the path on which the target pedestrians could be located, but less likely in the upper left or right corner for instance.

5. **Other characteristic features** (F_1 and F_7):

   - The number of segmented salient regions.
   - Average Euclidean distance between the centers of segmented salient regions and the centre of the image (F_7).

   These two characteristic features again capture the spatial distribution of the salient regions. Too many salient regions are not desirable and indicate presence of noise or distracting objects. Similarly, salient regions in close vicinity are more desirable as they might indicate different parts of the target object.

All the above-mentioned characteristic features are single valued except for the patch-based characteristic features where a characteristic feature is represented by a vector of length P. Hence, when concatenating all the characteristic feature values we get a characteristic feature vector of length 77 when nine patches are used.

6.4.3 The regression model

There exist numerous regression models that have shown great success in various fields. However, some of these state-of-the-art techniques have shown very promising results, particularly in active vision applications [155–157]. Random forest is one such powerful technique that gained popularity in many vision applications [158]. For some of the reasons provided in section 6.4.3.2, we have chosen random forest regression in this work.

6.4.3.1 Random forest regressor

Influenced by the early work of Amit and Geman on feature selection [159], Breiman's random forest technique emerged as one of the state-of-the-art techniques for handling a very large number of features [160]. Random forest is an ensemble based learning technique that utilizes the classification and regression tree (CART) structure [161] to produce a collection of random trees (called a random forest). Random trees are constructed in four steps:

1. Generate B bootstrap data by sampling a dataset independently with or without re-substitution. Each bootstrap sample will be used to construct a single tree. This choice of B has computational significance as too many trees requires more tree evaluation. However, according to the Strong Law of Large Numbers and tree structure, by increasing the number of trees, a limiting value of the generalization error is produced with little over-fitting [160].

2. For each bootstrap dataset, select a random set of features/variables indicated as m_{t} with the same distribution from D number of features such that $m_{\mathrm{t}} \ll D$. This is a key performance parameter for tuning random forest, which has more significance when the number of features is large.

3. Construct a deep CART tree using the above two parameters without pruning to a maximum tree size (i.e., a fixed terminal node size A). Growing a deep tree reduces the bias as more degrees of freedom are introduced.

4. Aggregate the results from individual ensembles (i.e., trees) to produce the final predicted value.

A single decision tree with L leaf nodes divides the feature space into L rectilinear regions in the feature space indicated by R_l where $1 \leq l \leq L$. For a given ensemble, the prediction model is given as:

$$h_j(\mathbf{x}, \mathbf{\Theta}_j) = \sum_{l=1}^{L} \Big(c_l \mathbb{1}(\mathbf{x}, R_l) \Big)_{\mathbf{\Theta}_j} \tag{6.4}$$

where $\mathbf{\Theta}_j$ is an independent identically distributed random vector that characterizes the j^{th} random forest tree using the splitting variable and the threshold cutting points at each node. The c_l values are computed by fitting a constant model independently at each region R_l. For regression this corresponds to the mean of the response variable y (i.e., AUC-ROC score) of samples belonging to region R_l. The indicator function $\mathbb{1}$ returns one if $\mathbf{x} \in R_l$ and zero otherwise.

Since the final prediction of a forest is the average of the ensemble predictions, the prediction model of the forest is given as:

$$H(\mathbf{x}) = \frac{1}{B} \Big(\sum_{j=1}^{B} h_j(\mathbf{x}, \mathbf{\Theta}_j) \Big) \tag{6.5}$$

Biau has shown theoretically that the main driving force behind the convergence of random forest technique is the strong set of features used during the learning phase [162]. The Random forest technique has the ability to identify such features by assigning an importance value to each feature. Out-of-bag (OOB) data (i.e., around 37% of the sample data used in each bootstrap) is used as validation set to compute the feature importance. The procedure for computing feature importance is as follows:

1. In a particular tree, a single out-of-bag sample is run down the tree and when it encounters a feature node split for which importance needs to be evaluated, a random branch is chosen. This process is repeated for all the out-of-bags samples and over all the trees for a particular feature.

2. The noisy prediction error PE_n for the feature computed over all out-of-bag examples. The whole process is repeated with the correct branching at the split point to estimate true prediction error PE.

3. Finally the importance of a feature F is given as:

$$\mathrm{Imp}_{(F)} = \mathrm{PE}_n - \mathrm{PE} \tag{6.6}$$

6.4.3.2 Why choose random forest?

The selection of a suitable regression model is dependent on the problem. Apart from being one of the state-of-the-art techniques, there are several reasons of selecting this model for our problem. The first is its ability to identify the important characteristic features of a feature map. This is particularly useful to establish a holistic description of the quality of a feature map. Secondly, tuning the model parameters is easy to perform as no separate test validation step is required. Instead, OOB error is estimated internally during the run.

Random forests are able to deal with unbalanced and missing data as our feature space has more AUC-ROC values greater than 0.5 than less than 0.5. In addition, when inspecting the characteristic features of the feature maps in the dataset, we have found that the data is noisy and scattered. The complexity of the data can be visualized in Fig. 6.7 for all three types of feature maps (i.e., BU, TM and CM). The characteristic features are extracted from the feature maps followed by a dimensionality reduction using *t*-Distributed Stochastic Neighbor Embedding (*t*-SNE) algorithm to a single projected dimension. This would roughly approximate the scattered nature of the data in a higher dimension.

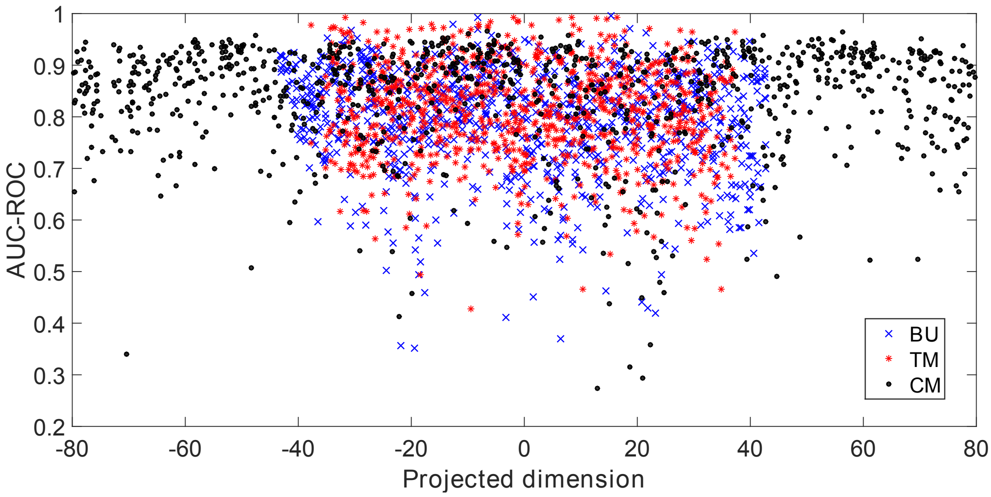


Figure 6.7: *D*-dimensions characteristic feature projection over a single dimension using *t*-SNE algorithm [163]. The projection is performed for the characteristic features of all three feature maps.

We would expect that a linear model would not perform well in this data. Normally, tree-based models in principle can fit data with any shape at the expense of over-fitting. In contrast, the random forest approach is able to discover more complex dependencies within the data with little over-fitting [160]. However, random forest may not be able to predict beyond the range of the independent variable values in the training data [162].

6.5 Feature map integration

In the previous section, we proposed a model that can estimate the quality score of any given novel feature map. To associate the model with an application, we have chosen the feature map integration framework proposed by Ehinger et al. [3] for predicting human fixation. As mentioned earlier in section 6.3, in this framework three different types of feature maps are generated. This is followed by integrating them to yield the final saliency map to predict where human fixate in images when searching for pedestrians.

Based on Ehinger's feature map integration framework, we propose two approaches for feature map integration that incorporate the feature map quality estimation procedure described in section 6.4. Both approaches are based on imposing a fixed threshold value for feature map selection.

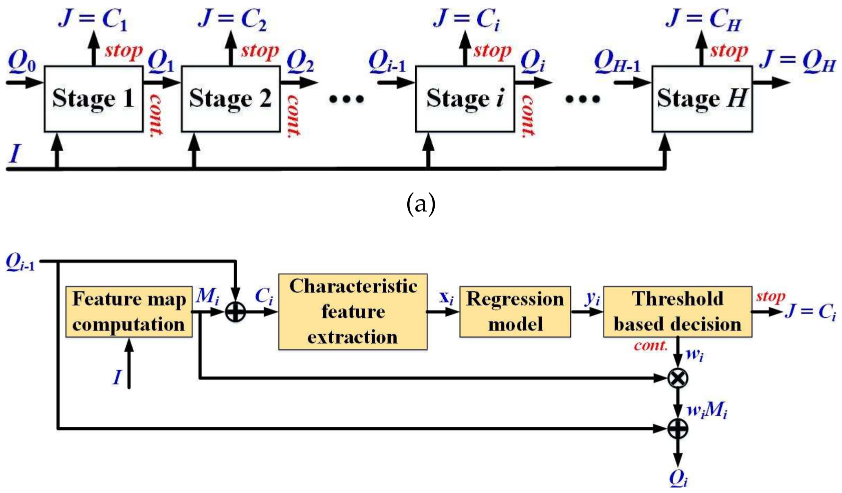
6.5.1 First approach: combined map with dual threshold (CMDT)

The model comprises of H iterative stages where each stage corresponds to a single feature map computation as illustrated in Fig. 6.8(a). The input to any stage i consists of the original image I and the output of the previous stage Q_{i-1} such that Q_0 is an initial map of all zero. In addition, upon achieving a 'stop' criterion, the iterative procedure stops and the output of the stage becomes the final output saliency map $J = C_i$, where C_i is some combination of maps performed in stage i. Otherwise, the procedure continues to produce an intermediate result (map) Q_i.

Figure 6.8(b) shows the detailed steps in a stage. The procedure starts by computing a feature map M_i through some technique denoted as 'feature map computation' i. From Ehinger's set of feature maps, this could be either a BU, TM or CM map generation technique. The next two modules belong to the feature map quality estimation model which extracts characteristic features from a map and feeds it to the regressor model to output an estimated AUC-ROC value y_i. A decision on the estimated value is performed as follows:

$$Action = \begin{cases} \textit{Stop and } J = C_i, & \text{if } y_i \geq t_s \\ \textit{Continue } w_i = 1, \text{o/p } = Q_i & \text{if } t_s > y_i \geq t_c \\ \textit{Continue } w_i = 0, \text{o/p } = Q_i & \text{otherwise} \end{cases} \tag{6.7}$$

where w_i is a weight assigned to the map if the '*continue*' criteria is selected.



(a)

(b)

Figure 6.8: The proposed combined map with dual threshold (CMDT) approach for feature map integration. The overall block diagram is shown in (a) for H stages. In this scenario, $H = 3$ as we have only three feature maps. The modules of a stage is shown in (b).

When the predicted value y_i exceeds an upper limit threshold value t_s, the procedure stops and the combined map for the stage becomes the final output map. However, when y_i is less than the lower threshold value t_c, the map is removed by assigning it a zero weight. Similarly, when y_i is between the two threshold values, it is retained. Ultimately the final output of a stage is the combination of the weighted map w_iM_i and the previous stage map Q_{i-1}.

It is obvious that the quality estimation model is applied to the combined feature map C_i and not directly on the computed feature map M_i. The combined feature map could belong to any of the possible $2^H - 1$ combinations. Hence, we train separate regression models for each combination. The reason for using a separate model for each combination is because the visual attributes of these maps are different (see Fig. 6.3).

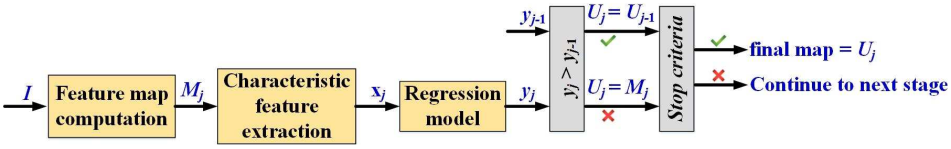


Figure 6.9: A single stage of winner map with single threshold (WMST) for feature map integration.

6.5.2 Second approach: winner map with single threshold (WMST)

In this approach, a single feature map is selected with the best estimated AUC-ROC value y_j to represent the final fixation map where j is the index of a stage. As shown in Fig. 6.9, at each stage, the estimated AUC-ROC value y_j is compared with the AUC-ROC value of the previous stage winner feature map U_{j-1}. The current feature map M_j becomes the winner feature map only if its estimated value exceeds y_{j-1}. If the estimated AUC-ROC value of the winner feature map exceeds a fixed threshold t_u, the process stops; otherwise it continues to the next stage.

6.6 Results and discussion

In this section, we discuss and analyze the results from three different perspectives. First, we examine how effectively our proposed approach estimates an AUC-ROC quality score of a feature map evaluated by the mean square error between the actual groundtruth score and the estimated score. Secondly, we see which characteristic features are important in each type of feature map. Finally, we illustrate how effective our approach is in the dynamic feature map integration process for predicting human fixation.

6.6.1 Properties of the dataset

The dataset we have used for all our experiments consists of 912 real world outdoor images of urban environments that is assembled by Ehinger et al. [3] for a visual search task. Half of the dataset contained the target pedestrian and the target was absent in the other half. In these colour images of dimension 600×800, on average the target roughly occupied a region of dimension 64×31. The background varies in complexity and consists of objects such as streets, cars, road signs and buildings. For the pedestrian-present images, the targets were spatially equally distributed across the images' four quadrants and at an angle range of $2.7^o - 13^o$ from the centre of the images [3].

After generating the three feature maps for all images, the AUC-ROC score is computed. It has been observed that the majority of these values have a moderate to high AUC-ROC value with only relatively few having low quality (below 0.5) in all three feature maps.

This suggests that the data is biased towards average to good maps with very few examples of low-quality maps. This may also suggest that the random forest regressor is very likely not to be trained to a range of values of the characteristic features from the low-quality feature maps. As a result, generalizing to cases with completely new data from low-quality feature maps would be problematic. Hence, we might expect the regression model to exhibit poor estimation performance on low-quality feature maps.

6.6.2 Simulation and parameter setting

From the outdoor images dataset assembled by Ehinger et al., 100 images were used to train the pedestrian detector and another 100 to train the contextual model. None of these images were used to train or test our model. The remaining 712 images were divided equally for the training and testing phases. We used the online code by Ehinger et al. for generating the three feature maps without altering the parameters for map generation [164].

In order to select the best parameters for the random forest regressor, we empirically tested the regression model over a grid of m_{t} and B values. We have noticed no significant improvement in the out-of-bag prediction error beyond $10,000$ trees in any of the three feature maps. Hence in all our experiments, B is fixed at $10,000$ trees. For m_{t}, Table 6.1 shows the best-selected values for all three feature maps. In addition, each experiment is conducted 30 times. Along with the mean results, the margin of error with confidence interval of 95% is reported. Table 6.1 lists the parameter values used in our experiments.

All the experiments are conducted on a single *Intel core i7-4790 @* 3.60*GHz* machine with 8Gb memory and running the Linux operating system. The simulations are performed using MATLAB R2015b.

6.6.3 AUC-ROC estimation performance

To find the performance of our proposed approach in estimating the AUC-ROC score, we report the average mean-square-error (MSE) and the margin of error for a 95% confidence interval between the estimated AUC-ROC scores and the groundtruth scores of the testing images over 30 runs per experiment.

6.6.3.1 Feature map analysis

Table 6.2 shows the achieved MSE results for different feature maps. The $'+$ indicates the combination of maps. Note that for notation, the term to the left of $'/'$ represents the training feature map type whereas to the right is the testing feature map type.

From the table it can be observed that the performance varies as the training and testing feature maps vary. When the training and testing feature maps are similar, our proposed approach performed very well and

Table 6.1: Parameter values used in the experiments.

Description	**Notation**	**Value**
Characteristic features		
# of characteristic features	R	29
characteristic feature vector length	D	77
# of patches	P	9
Random forest regressor		
# of random trees	B	$10,000$
# of tree characteristic features	m_t	BU:5,TM:10,CM:8
terminal node size	A	5
Bootstrap sample size	O	$0.632 \times N = 225$
Bootstrap re-substitution	-	Applied
Miscellaneous		
# of experiments	-	30
# of train/test images	N	356 each
Fixation threshold	K	10%
# of feature maps	H	3 (BU, TM and CM)

the MSE is lowest (see the diagonal values of the table). Small MSE indicates that the regression model was able to estimate the quality of a feature map accurately.

For instance, when the regression model is trained and tested over (BU+TM) feature maps (we denote it as (BU+TM)/(BU+TM)), an MSE of 0.0063 is achieved. On the other hand, when testing on different feature maps (e.g., all MSE values in the fourth column of the table other than BU+TM), the MSE values increase due to the difference in visual attributes between the training and testing feature maps. The same can be observed in all the columns of the table.

As discussed in section 6.3, the visual appearance of the three types of feature maps (i.e., BU, TM and CM) are different. The extracted characteristic features from the features maps form certain patterns in a high dimensional space that vary with the type of feature map. As a result, when trained on one type and tested on another, the estimation performance is degraded. In a similar manner, when two or more feature maps are combined, the resultant feature map has different visual attributes than the feature maps that contribute in its combination.

For instance in Table 6.2, when training over the composite feature map (BU+TM), the average MSE is 0.0094 and 0.0236 for testing over BU and TM respectively. Both values are high compared to that achieved by testing over (BU+TM) feature map (i.e., 0.0063). This is because the combined feature map exhibits different visual attributes from those feature maps used in constructing the combined map. In fact, it can be typically observed from the table that the characteristic features extracted from the composite feature maps are better in describing the map's visual attributes than those extracted from single feature maps.

For a single trained and tested feature maps (i.e., BU/BU, TM/TM and CM/CM), the average MSE achieved by the proposed approach is 0.0072, 0.0060 and 0.0087 for BU, TM and CM feature maps respectively. In contrast, for the composite feature maps (see (BU+TM/BU+TM), (BU+CM/BU+CM), (TM+CM/TM+CM) and (BU+TM+CM/BU+TM+CM)), the MSE is normally lower than those for single feature maps. The best result of 0.0044 is achieved when all feature maps are combined. This suggests that when feature maps are combined, the characteristic features become more effective in describing the maps.

According to the result presented in Table 6.2, it is clear that the combination of the extracted characteristic features and the random forest regressor was able to estimate the quality score of a novel feature map accurately under specific training and testing limitations. This limitation restricts the feature maps to be of similar type in both phases. If this condition is satisfied, our proposed model can estimate a quality score of any feature map with high precision.

Table 6.2: The MSE quality estimation performance for different feature maps. The ’All’ feature maps refer to a combination of BU, TM and CM or can be denoted as (BU+TM+CM). The average MSE is calculated over 30 runs. The margin-of-error is computed from the standard error in the sample mean and expanded to a confidence interval of 95%.

		Training maps						
		BU	TM	CM	(BU+TM)	(BU+CM)	(TM+CM)	All
Testing maps	BU	0.0072 ± 0.00008	0.0105 ± 0.00027	0.0120 ± 0.00070	0.0094 ± 0.00022	0.0097 ± 0.00034	0.0159 ± 0.00185	0.0106 ± 0.00030
	TM	0.0228 ± 0.00170	0.0060 ± 0.00008	0.0333 ± 0.00500	0.0236 ± 0.00253	0.0143 ± 0.00131	0.0390 ± 0.00450	0.0163 ± 0.00112
	CM	0.0101 ± 0.00036	0.0122 ± 0.00072	0.0087 ± 0.00013	0.0114 ± 0.00054	0.0098 ± 0.00047	0.0117 ± 0.00030	0.0108 ± 0.00035
	(BU+TM)	0.0223 ± 0.00170	0.0095 ± 0.00029	0.0202 ± 0.02401	0.0063 ± 0.00020	0.0129 ± 0.00092	0.0202 ± 0.00195	0.0103 ± 0.00050
	(BU+CM)	0.0113 ± 0.00043	0.0100 ± 0.00064	0.0277 ± 0.00172	0.0110 ± 0.00073	0.0048 ± 0.00015	0.0289 ± 0.00310	0.0066 ± 0.00019
	(TM+CM)	0.0247 ± 0.00120	0.0112 ± 0.00048	0.0304 ± 0.00274	0.0111 ± 0.00005	0.0130 ± 0.00067	0.0079 ± 0.00025	0.0090 ± 0.00028
	All	0.0307 ± 0.00170	0.0105 ± 0.00076	0.0447 ± 0.00349	0.0093 ± 0.00040	0.0128 ± 0.00073	0.0305 ± 0.00372	0.0044 ± 0.00013

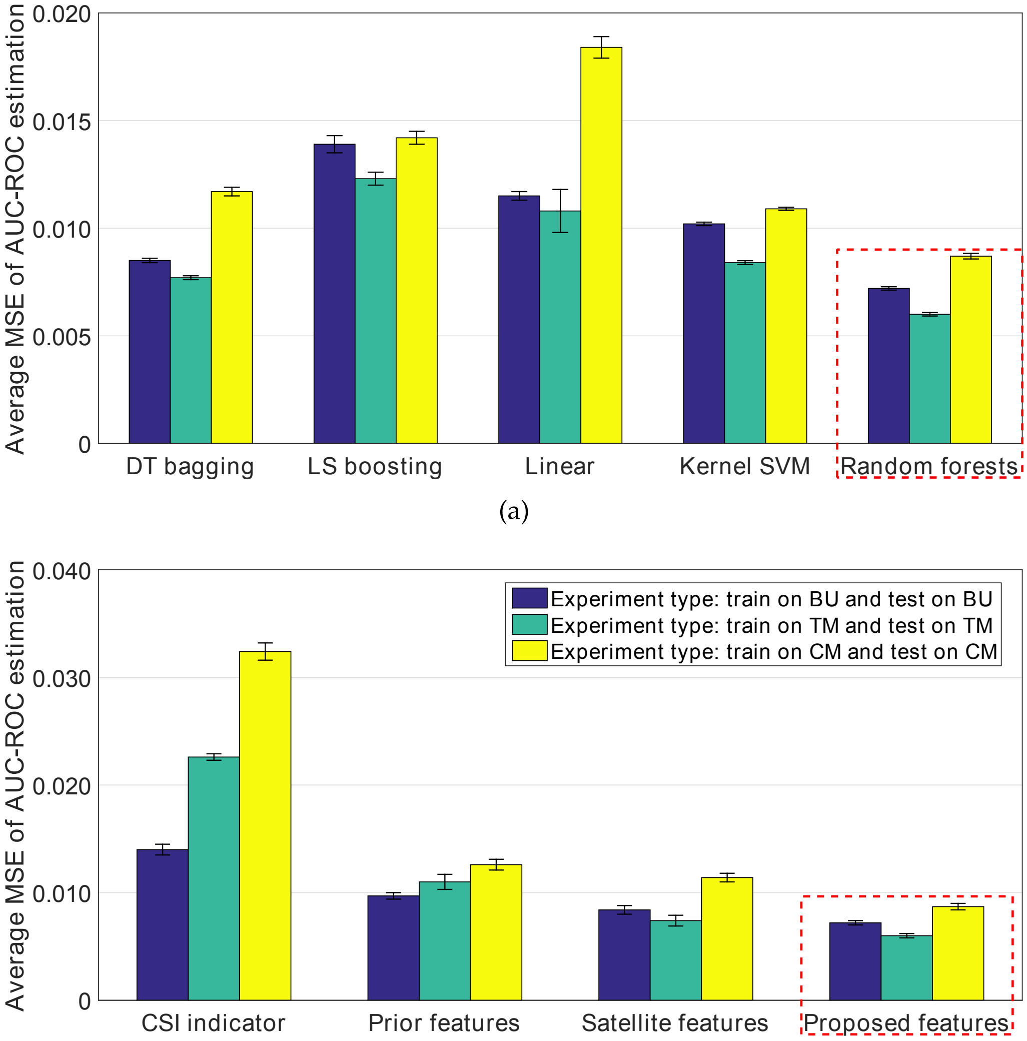
6.6.3.2 Estimation performance comparison

In the previous section, we have seen that our model accurately estimates the AUC-ROC score of a given feature map. To justify our selection of the characteristic features and the random forest regressor, in this section we compare the average MSE achieved by our selection with other possible choices and show that our model is a suitable choice for the estimation problem.

We have compared the random forest regressor with the following alternative regression techniques: decision tree bagging [165], least square boosting (LS Boost) [166], a fitted linear model, and Kernel-SVM. The comparison is performed over BU/BU, TM/TM and CM/CM cases. As shown in Fig. 6.10(a), random forest regressor outperforms all other techniques in all three cases. Perhaps the most competitive technique to the random forest is tree bagging. This is expected as bagging is similar to random forest except that no feature sampling is performed in the former. As a result, some level of variance exists in the generated trees and is reflected by the MSE values when compared with those of random forests.

To validate the effectiveness of the characteristic features we used (see the 29 set of features described in section 6.4.2), we compare its performance with some other set of characteristic features proposed previously for saliency map quality estimation. Specifically, we compare the performance of our characteristic features with Kim et al. features (we call these *prior features*) [32], Hu et al. CSI indicator [109] and the features for satellite images (*satellite features*) proposed by Li and Itti [12]. For a fair comparison, all these features are used with random forest regressor.

As shown in Fig. 6.10(b), our characteristic features (we refer to as the proposed features in the figure) achieved the smallest mean square error in all the feature maps. In contrast, the CSI indicator performed poorly in estimating the AUC-ROC scores. Since the CSI indicator concretely relies on the compactness characteristic feature which is less effective in fixation maps, it gives high MSE values. *Prior features* is more effective than



(a)

(b)

Figure 6.10: Average mean-square-error (MSE) of the proposed model using random forest regressor. The error bars represent the margin-of-error with confidence interval of 95%. The error is computed when the model is trained and tested on the same type of feature maps. The performance of our model is compared with other regression models shown in (a). In addition, when the random forest regressor is trained with different set of characteristic features, our proposed characteristic features has the smallest MSE in all three feature maps as shown in (b).

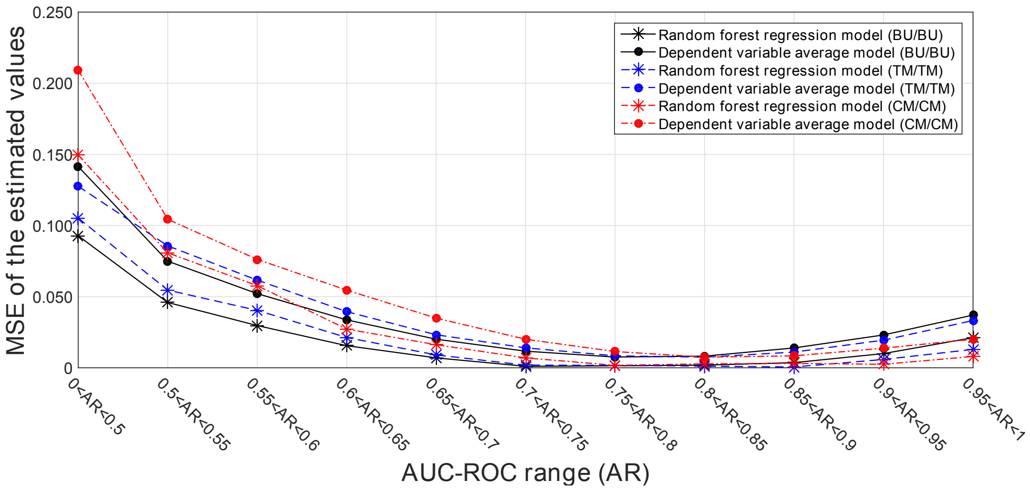


Figure 6.11: The MSE quality score estimation performance comparison between random forest regressor and an average model over a range of AUC-ROC values for BU, TM and CM feature maps.

CSI indicator, despite using compactness because it also uses some other useful characteristic features such as brightness and darkness priors. The characteristic features proposed by Li and Itti for satellite images have a comparable performance to our characteristic features in all three feature maps. However, these characteristic features are computationally expensive particularly in large images.

Because random forest performs an averaging operation (i.e., fitting a constant model) within a rectilinear region in the feature space, it would be interesting to compare its performance against a simple average model. Figure 6.11 shows the break-up of MSE of estimation at various AUC-ROC splits for both random forest regressor and an average model. The average model simply computes a quality score by averaging the AUC-ROC values over all the examples.

In all three feature maps, it is evident that random forest is far better than a simple average model in estimating an AUC-ROC value. This performance varies from one split to another. Because the majority of the feature maps in all three cases have AUC-ROC values greater than 0.7, the average model performs slightly better within this range than the range less than 0.7.

The results presented in this section confirms the viability of the random forest regression techniques and the set of characteristic features for the type of feature maps used in our model.

6.6.3.3 Visual results

Figure 6.12 shows three types of visual examples (i.e., best, worst and a typical estimated AUC-ROC score) from BU/BU, TM/TM and CM/CM cases respectively. For the best performance achieved in both TM and BU maps, the absolute difference between the actual groundtruth value (in green) and the estimated value (in red) is around 0.006 and 0.007 respectively, indicating highly accurate estimation. For the CM feature map, this difference is also low (around 0.023). However, in most typical maps (see the sample image selected as an example), the CM feature map error is around 0.089. As the MSE for BU and TM feature maps reported in the previous paragraphs is lower than that of CM, a small difference between the actual and the estimated value can be observed from the typical sample images for BU and TM feature maps.

6.6.4 Feature importance

As mentioned in section 6.4.3.1, the random forest technique has the ability to determine the importance of each characteristic feature in the regression problem using Eq. (6.6). Figure 6.13 gives a complete importance profile for BU/BU, TM/TM and CM/CM measured using a decrease in the Gini index value. This value is directly proportional to the importance of the characteristic feature.

For both BU and TM feature maps, the most important characteristic feature is the global mean intensity of the map (f_{73}). This characteristic feature describes the overall brightness of a feature map and it is one of the

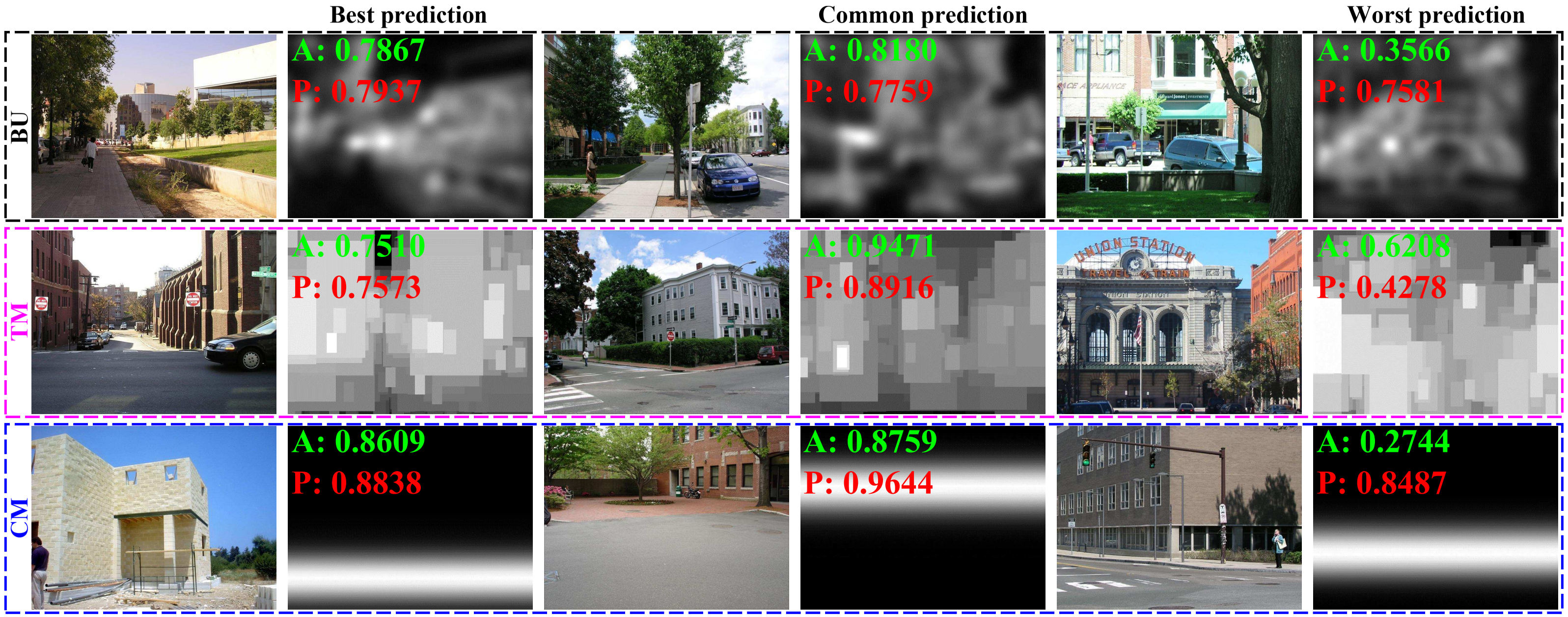


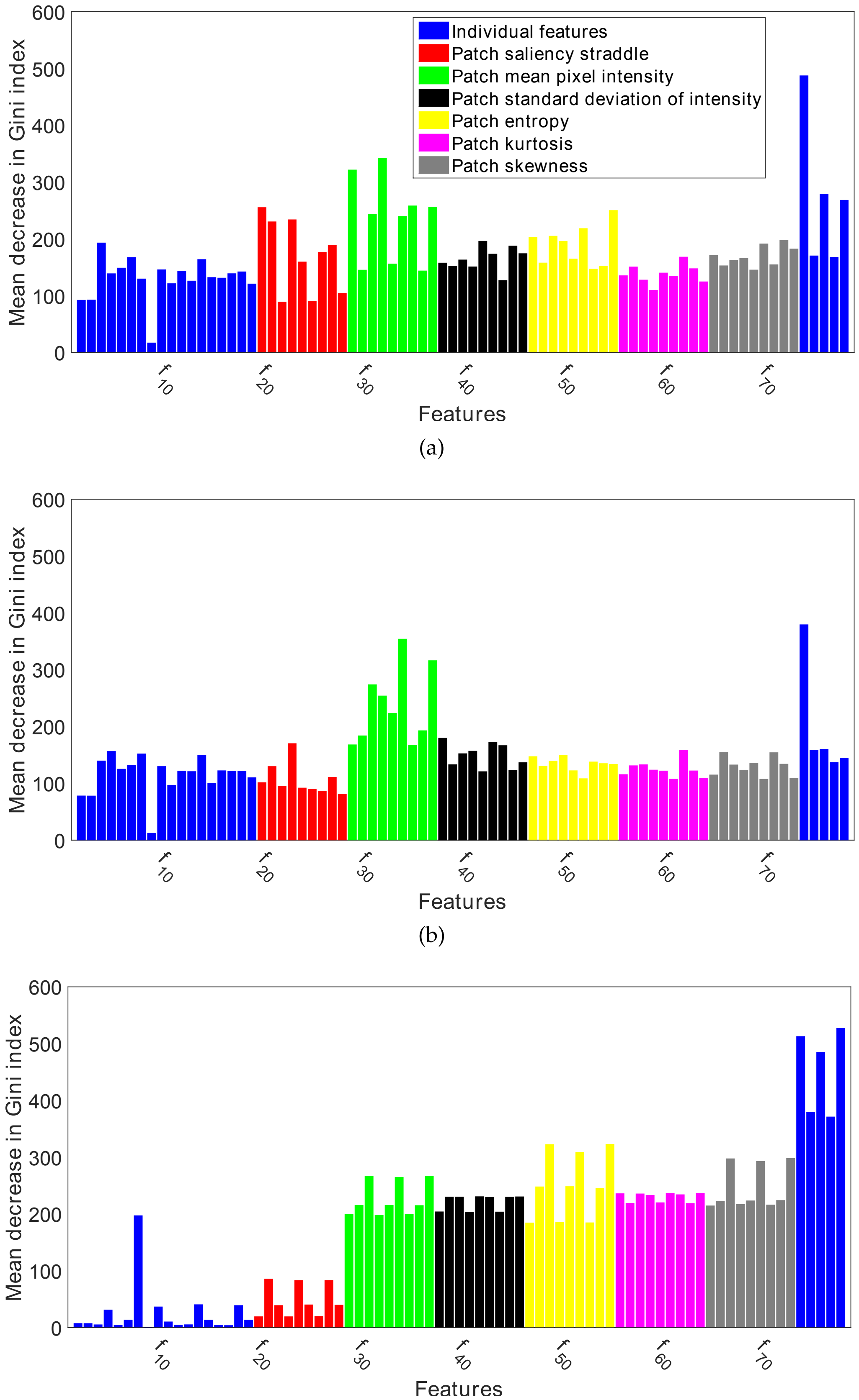
Figure 6.12: Sample images from the best, typical and worst estimated AUC-ROC values by the proposed approach. The value in green represents the actual groundtruth value of a feature map (denoted by *'A'*) whereas the estimated value *'P'* is in red.

main characteristic features used by Kim et al. in the *prior features*) [32]. In addition, the global entropy (f_{75}) of the BU feature map is also important as it captures the scatteredness of the fixation map. In contrast, the TM feature map is inherently scattered due to the way these maps are generated [62], and hence, it does not receive high importance. All single valued individual characteristic features have moderate importance in both the feature maps except for the global density which is least important. Again this characteristic feature is taken from CSI indicator and has more significance in salient object maps than for fixation or TM feature maps.

Patch based characteristic features have almost uniform importance over all the patches in both the feature maps. Some exceptions suggest that certain patches are more important than other patches, for instance, the patch saliency straddle for BU feature maps. A particular importance pattern can be observed (i.e., patches 1,2,4,5,7 and 8) which indicates that the upper half of the maps are more dense in salient regions than the lower part, and hence more important. Roughly, this gives an overall picture of the regions of interest highlighted by the BU feature maps.

It can be seen from Fig. 6.13 that the feature importance of BU characteristic features is highly correlated with that of TM. Despite this similarity, the estimation performance for TM feature maps is better than BU feature maps as discussed in the previous section (see Table 6.2). This is due to the way these characteristic features behave in high dimension and due to their level of dependency that varies from a feature map to another when learning an estimation model.

For the CM feature maps, local patch-based characteristic features have received high importance by random forest regressor along with the global statistical characteristic features ($f_{73} - f_{77}$). The contextual feature map is visually a strip of uniform intensity, so the local characteristic features capture this spatial distribution. Furthermore, in contrast to BU feature maps, the pattern of the patch importance suggests that the lower part of the map is more important. Similarly, the distance from the center of



(a)

(b)

(c)

Figure 6.13: Characteristic feature importance profile determined by random forest regressor for (a) BU/BU feature maps, (b) TM/TM feature maps and (c) CM/CM feature maps. In order to interpret this figure, it should be viewed in colour.

the map (i.e., f_7) is also given an importance because it provides location information of the salient strip. Note that no shape based characteristic features (i.e., geometrical) are given importance as such information does not matter when considering a single rectangular horizontal strip.

Note that some of the characteristic features discussed in section 6.4.2 would be redundant in some feature maps (i.e., exhibiting low importance) and relatively important in other type of feature maps. For instance, $f_1 - f_{28}$ have least importance in CM/CM but exhibit more importance in both BU/BU and TM/TM cases (see Fig. 6.13).

Typically, for important characteristic features, we expect that such characteristic features when considered solely would have comparable estimation accuracy to that achieved when all features are considered. To elaborate this point through example, we visualize the impact of characteristic feature importance on the MSE estimation performance by fitting two regression models on the data. The first model is learned when only global mean intensity (f_{73}) is extracted from all the BU feature maps whereas the second model is learned from the CM feature maps when using normalized size of LSR (f_{10}). As shown in Fig. 6.14, when only using f_{73}, a better fit to the data is achieved than f_{10}. The MSE between the fitted model and the groundtruth AUC-ROC values is 0.0113. Since f_{73} is the most important characteristic feature in BU feature maps, the achieved MSE of estimation is close to that achieved when all 77 characteristic features are used (i.e., 0.0072).

In contrast, f_{10} in CM feature maps has a very low importance which is can be visualized by the fitted model (see Fig. 6.14(b)). The majority of the values for this characteristic feature spans a very small range which is centralized around 0.1 with the corresponding AUC-ROC values ranging from 0.35 to 0.93. Only some examples have their values outside this range. As a result, by fitting a model that is solely learned from this characteristic feature on the data, the MSE of estimation is very high (i.e., 0.0324 compared to 0.0087 when all the characteristic features are used). This

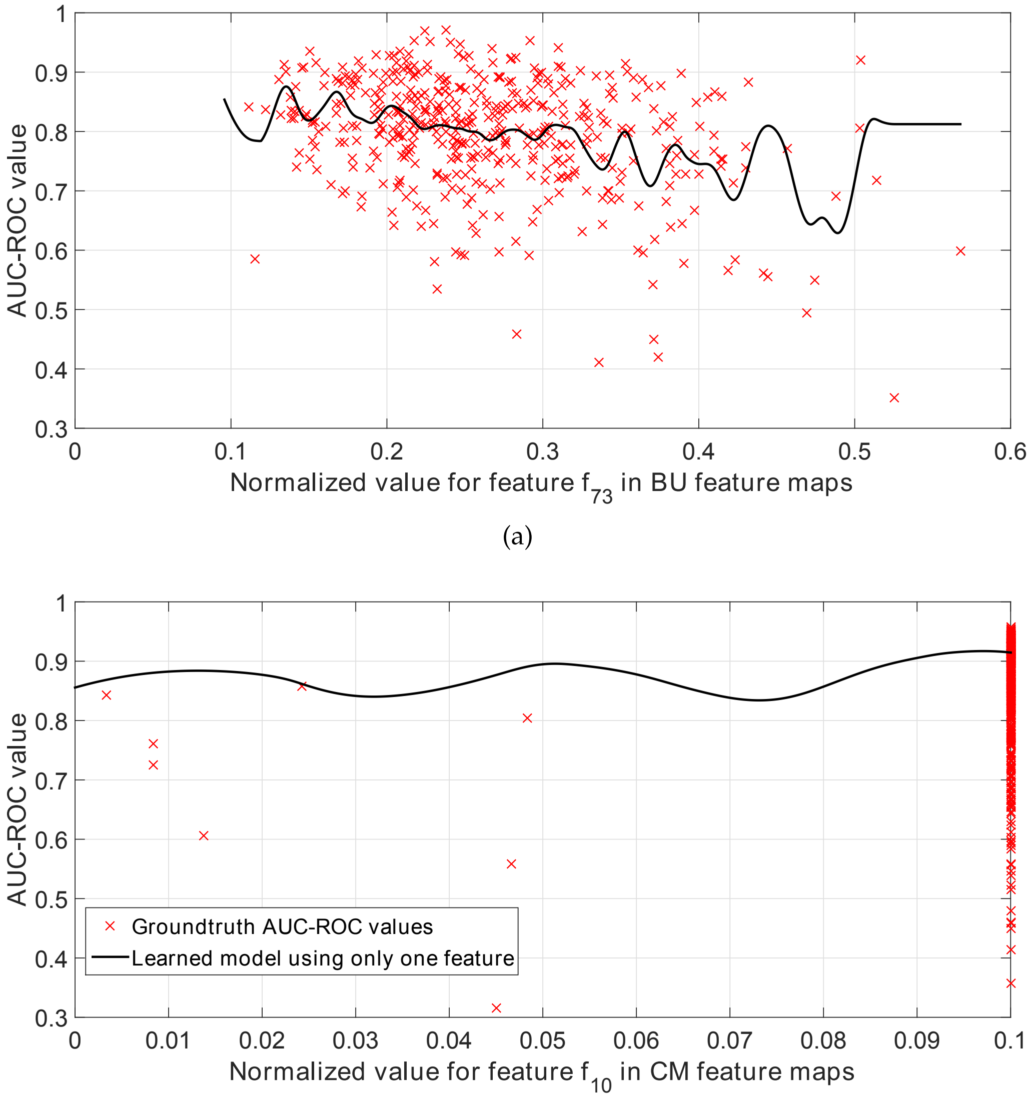


Figure 6.14: Model fitting on the data when learned from (a) An important BU feature map characteristic feature f_{73} and (b) f_{10} from CM feature map with low importance.

means that for the CM feature map, the normalized size of LSR (f_{10}) does not capture any structure with the AUC-ROC and would result in poor estimation performance if considered alone.

The important characteristic feature profile provides a way to select those characteristic features that would be most useful in estimating a quality score of a feature map. These characteristic features vary in importance from one feature map to another. Note that such feature importance does not consider the interaction between features and only performs the selection based on individual performance. Ideally, we would perform a feature selection over the set of characteristic features to acquire a set of selected characteristic feature to maximize both accuracy and efficiency. However, this is not considered in this work and can be allocated for future work.

6.6.5 Characteristic feature analysis

In this section, we investigate the nature of some of the characteristic features and their relation with the estimated AUC-ROC values. This is achieved by producing partial dependency plots of characteristic features. These plots are generated by assigning various possible values to the candidate characteristic feature and averaging the response from all the trees in the random forests over all the examples. We haven chosen six interesting characteristic features from each feature map for discussion.

The characteristic features from left to right in Fig. 6.15 are $f_1, f_2, f_{15}, f_{73}, f_{74}$ and f_{75} respectively for BU/BU (i.e., when train on BU and test on BU feature maps) (see first row), TM/TM (second row) and CM/CM (third row).

The first characteristic feature is the number of salient regions in a feature map. Normally having more salient regions is not a desirable visual attribute in a feature map as it accounts for more regions of interest. This behavior is clearly captured by the dependency plot for BU feature map. For TM feature maps, no obvious relation can be deduced. This might

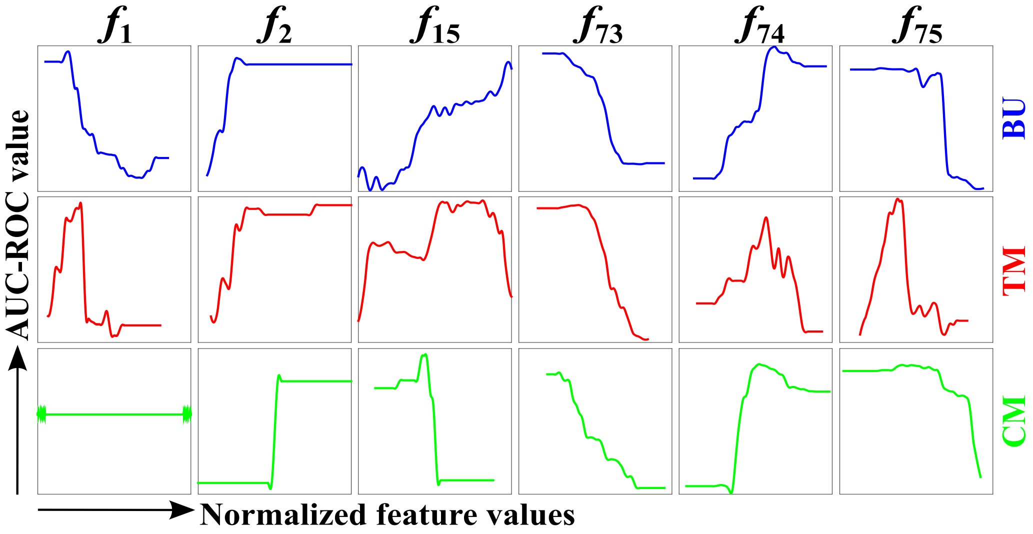


Figure 6.15: Partial dependency plots of sample characteristic features for BU (first row), TM (second row) and CM (third row) feature maps.

be due to the procedure by which such target maps are constructed (i.e., upright rectangular shaped regions). For CM feature maps, since we are dealing with a single strip of region, the relation is simply a constant. This explains why this characteristic feature has low importance in the CM feature maps.

In the next characteristic feature which represents the average size of the salient regions (f_2), all the dependency plots show that larger salient regions in a feature map indicate a better quality map than smaller regions. Moving to the next characteristic feature, which is the normalized orientation of the most salient region (f_{15}), the normalizing is performed such that the value of one represents $90°$ orientation and -1 is $-90°$. However, the partial dependency plot for this characteristic feature shows the feature values ranging from zero to one (i.e., the x-axis).

Although this characteristic feature has low importance, it has some interesting information. Both BU and TM maps show that a map of its most salient region having an orientation close to $90°$ is more desirable. This observation is in an accordance with the visual task of finding pedestrians that usually are found in an upright position in natural scenes. In contrast, the CM feature maps show a different preference of horizontal orientation. This is not surprising as most of the contextual regions (mostly roads or pavements) in this dataset are located horizontally.

In all feature maps, for the saliency global intensity characteristic feature (f_{73}), a nearly linear relation exists between the values of this characteristic feature and the estimated AUC-ROC values. The average intensity of a feature map corresponds to the brightness of the map. Good feature map exhibits low overall brightness (see the partial dependency plot for (f_{73})) and high contrast (captured by standard deviation of the feature map (f_{74})). High contrast feature map yields high AUC-ROC value which is again captured by the partial dependency plots for f_{74}. Finally, the global entropy f_{75} measures the randomness of a feature map. The partial plots show that higher the entropy, lower becomes the quality of the feature map. Hence, a good feature map is characterized by low saliency randomness.

The above examples show that we can deduce a holistic description of the quality of a feature map from the individual characteristic features through the partial dependency plots.

6.6.6 Feature map integration performance

To evaluate the performance of our feature map integration approaches, we present the final AUC-ROC value achieved when combining the feature maps used the proposed CMDT and WMST combination approaches. This AUC-ROC value will evaluate the overall system's performance for predicting human fixation. We show that both approaches yield higher accuracy in prediction human fixation and better efficiency than that achieved by combining all feature maps.

6.6.6.1 CMDT performance

In CMDT approach, a key factor in performing the decision making on the estimated value by the regressor is the selection of the upper t_s and lower t_c threshold values. We test CMDT on a grid of different threshold values between 0.5 and 0.95. Values less than 0.5 were not applied as they correspond to chance (or worse). For each threshold pair, we compute the average AUC-ROC of the final fixation map generated by CMDT for all the test images. Further averaging was performed over 30 experiments for repeatability.

A performance improvement is observed when higher threshold values are used. Higher threshold values impose a constraint over premature stopping and results in higher AUC-ROC values. However, this is not the only factor that regulates the performance of the system. If the quality estimation model exhibits a high average MSE, then this will cause a wrong estimation of the AUC-ROC values at each stage of the CMDT procedure. This leads to a random error propagating from one stage to another. Even if an optimum set of thresholds is used, the decision will be made on false estimated AUC-ROC values.

To analyze the performance of CMDT, we have selected two points on the grid, the first having $t_c = 0.7$ and $t_c = 0.9$ and the second with $t_c = 0.7$ and $t_c = 0.8$. The corresponding final average AUC-ROC prediction value is 0.8867 and 0.8703 respectively. The average MSE over all the stages and for all the test images is 0.0058 and 0.0069 respectively. Furthermore, the average processing time of a single image through the CMDT stages takes approximately 4.03 and 3.85 seconds respectively. Hence, a trade-off exists between the two points. Higher prediction accuracy is achieved at the expense of more processing time.

From the results obtained through these two points, we can assert that as MSE is decreased (i.e., better quality score estimation), the overall average AUC-ROC value of the final fixation map is improved. Hence, the overall performance of CMDT is dependent on the threshold values as

Table 6.3: The impact of quality estimation performance on the overall performance of CMDT which is evaluated using the average AUC-ROC value and the average processing time.

Method	**Avg. MSE**	**Avg. AUC-ROC**	**Avg. time**
Random forest	0.0069 ± 0.0004	0.8703 ± 0.026	3.85 ± 0.07
DT Bagging	0.0093 ± 0.0006	0.8171 ± 0.017	3.97 ± 0.06
LS-boost	0.0121 ± 0.0004	0.7828 ± 0.028	4.01 ± 0.08
Linear	0.0148 ± 0.0005	0.7634 ± 0.031	4.03 ± 0.04

well as the performance of the regression model.

In order to see the impact of the estimation process on the overall performance of CMDT, we fix the thresholds to ($t_s = 0.8, t_c = 0.7$) to rule out its contribution and vary the regression model. Apart from random forest regressor, we use decision tree bagging [165], least square boosting (LS Boost) [166] and a linear model. Table 6.3 shows that the performance of a regression model for estimating a quality score is directly proportional to the overall fixation prediction accuracy. In addition, as the MSE goes high, decision is made on false estimated values. As a result, more stages are required to reach the stopping criteria, which requires increased processing time.

6.6.6.2 WMST performance

As in CMDT approach, various threshold values were used to evaluate the performance of WMST. For instance, when fixing the value of t_u to 0.84, an AUC-ROC of 0.8640 was achieved. When increasing the threshold to 0.9, even higher human fixation prediction was achieved (AUC-ROC of

Table 6.4: The importance of feature map computation sequence in improving the efficiency of WMST approach. The threshold value is fixed to $t_u = 0.9$.

Sequence of source	**Average AUC-ROC**	**Processing time (s)**
BU, TM, CM	0.8771 ± 0.014	4.03 ± 0.01
BU, CM, TM	0.8765 ± 0.015	4.01 ± 0.02
TM, BU, CM	0.8770 ± 0.012	3.74 ± 0.04
TM, CM, BU	0.8767 ± 0.015	3.14 ± 0.04
CM, BU, TM	0.8765 ± 0.017	3.37 ± 0.05
CM, TM, BU	0.8769 ± 0.013	3.12 ± 0.04

0.8771). Note that WMST operates on a single winner map that has an estimated AUC-ROC value exceeding a threshold value. Thus, the sequence of computing the feature maps impacts the efficiency of the approach.

Table 6.4 shows how changing the sequence of feature map computation can have an impact on the efficiency of the process. A considerable decrease in processing time from 4.03 seconds when the map manipulation sequence is (BU,TM,CM) to only 3.12 seconds with unnoticeable degradation in accuracy performance by reversing the order.

6.6.6.3 Performance comparison with various approaches

Table 6.5 compares the performance of the proposed approaches with various other approaches. The terms 'preferred' and 'best' refer to the performance when using $t_c = 0.7, t_c = 0.8$ and $t_c = 0.7, t_c = 0.9$ in the context of

CMDT respectively. For WMST, they correspond to the performance when $t_u = 0.9$ for the sequence 'BU,TM,CM' and 'CM,TM,BU' respectively.

When all feature maps are used (i.e., Ehinger's approach), the average AUC-ROC is 0.8524 and the time required to compute all the feature maps is 4.02 seconds. Our proposed CMDT and WMST approaches (whether it is 'preferred' or 'best') achieve higher fixation prediction accuracy. The 'best' approaches in CMDT requires slightly longer processing time, otherwise, the processing time is lower than computing all feature maps. The ideal case represents the best possible combination of feature maps that achieves the highest AUC-ROC value on individual images. Our proposed approaches, particularly CMDT (best) has a comparable prediction accuracy result with that of the ideal case. Note that the time required for estimating the quality score of a single feature map is only 0.01 second.

Finally, in order to demonstrate that the quality of the learned regression model has a major impact on the dynamic feature map selection process and hence, in human fixation prediction accuracy, we used other regression models with the WMST framework and reported the achieved AUC-ROC values. It is clearly evident that as the average MSE of estimation over all three stages increases, the AUC-ROC fixation prediction value decreases (see Table 6.6).

6.7 Chapter summary

One of the challenging tasks in visual attention systems is to identify the quality of a novel feature map based on its visual appearance. This process is particularly useful for integrating multiple feature maps of good quality to maximize the detection accuracy of a target object. Most of the previous approaches use compactness as the main parameter for quality evaluation. Compactness can be useful for evaluating the quality of feature maps when performing salient object detection but fail to describe the characteristic of fixation maps. In addition, these approaches address

Table 6.5: Human fixation prediction and processing time comparison between the proposed feature map integration approaches and other static combination of feature maps. τ is a very small value due to variation in machine speed.

Method	**Avg. AUC-ROC**	**Avg. time**
Only BU	0.7933 ± 0.014	$3.70 \pm \tau$
Only TM	0.8094 ± 0.020	$0.18 \pm \tau$
Only CM	0.8415 ± 0.018	$0.14 \pm \tau$
BU + TM	0.8042 ± 0.023	$3.88 \pm \tau$
BU + CM	0.8414 ± 0.019	$3.84 \pm \tau$
TM + CM	0.8436 ± 0.024	$0.32 \pm \tau$
All sources	0.8524 ± 0.027	$4.02 \pm \tau$
Ideal	0.8986 ± 0.020	3.97 ± 0.09
CMDT (preferred)	0.8703 ± 0.026	3.85 ± 0.07
CMDT (best)	0.8867 ± 0.021	4.03 ± 0.02
WMST (preferred)	0.8769 ± 0.013	3.12 ± 0.04
WMST (best)	0.8771 ± 0.014	4.03 ± 0.04

Table 6.6: The impact of feature map quality estimation on the overall performance of WMST through the average AUC-ROC value. The threshold value t_u is fixed to 0.84 for all the experiments. The margin-of-error is computed from the standard error of sample mean and t-distribution critical value with confidence interval of 95%.

Method	**Avg. MSE**	**Avg. AUC-ROC**
Random forests	0.0068 ± 0.00013	0.8640 ± 0.012
DT Bagging	0.0092 ± 0.00024	0.8129 ± 0.023
Kernel SVM	0.0113 ± 0.00009	0.7903 ± 0.012
Linear	0.0120 ± 0.00089	0.7754 ± 0.018
LS-boost	0.0131 ± 0.00050	0.7553 ± 0.027

the problem as a binary classification based on some criteria and do not provide an actual estimate of the quality measure.

In this chapter, we have proposed an approach that estimates a quality score of any novel feature map for fixation feature maps. The novelty of the proposed approach is twofold. The first is the proposed set of simple, efficient and effective features referred to as *characteristic features* that are extracted from a feature map to attain a visual description of the map. The second is to use these characteristic features to learn a regression model using random forests that estimates a quality score (i.e., AUC-ROC value) of any novel feature map. Targeting one particular dataset of 900 outdoor pedestrian images [3], we show that our approach has estimated the AUC-ROC score of three types of feature maps (i.e., bottom-up (BU), target map (TM) and contextual map (CM)) with high accuracy.

The proposed characteristic features provide a better description of the feature maps than three state-of-the-art previously proposed features used as feature map visual descriptors. By applying the random forest regression model on these three features, we found that the mean-square-error (MSE) of AUC-ROC estimation is higher than that achieved when using our proposed characteristic features on all three types of maps.

When training and testing the regression model over similar type of feature maps, our approach achieved the highest accuracy in estimating the AUC-ROC quality score. The effectiveness of the characteristic features is highly coupled with the structure of these features in the high dimensional feature space. Some of these complex relations are captured by the feature importance and the partial dependency plots. From this information, it is concluded that the proposed characteristic features, particularly the important ones in each type of feature map, can provide a rough visualization of a good feature map for fixation.

Finally, by integrating our proposed approach with a two simple proposed feature map integration frameworks, we demonstrated that our model achieved a higher human fixation prediction accuracy than that achieved by combining all feature maps adopted by Ehinger et al. [3]. In the first approach called the combined map with dual threshold (CMDT), the best average AUC-ROC prediction value is 0.8867 when the upper and lower threshold values are 0.7 and 0.9 respectively. Similarly, in the second approach, called winner map with single threshold (WMST) the best result is 0.8771 when the threshold value is 0.9. In both cases, the fixation prediction accuracy of the proposed approaches is higher than that achieved by integrating all feature maps which is 0.8524 at the expense of a minor increase in the processing time.

Chapter 7

Conclusions and Future Work

The primary goal of this thesis was to model top-down saliency and to combine it with bottom-up saliency to maximize the detection accuracy of an object detection task. This goal was successfully achieved by proposing a number of new approaches and methods for attentional feature weighting, feature selection and dynamic combination of bottom-up and top-down saliencies. The effectiveness of these methods was evaluated with two criteria; the accuracy of detecting a variety of target objects and the efficiency of the methods for practical purposes.

All these methods were evaluated using standard benchmark datasets for salient object detection and object recognition. In addition, a number of self-created challenging datasets for target object detection were used. The reason for having such datasets was to investigate the ability of the proposed methods to handle target objects with varying degree of saliency, which could not be performed on the available benchmark datasets.

In this thesis, four main contributions (in the form of proposed methods) are made in modeling task-driven visual attention. The first contribution is modelling of top-down saliency through feature weighting mechanism using contextual information. The second contribution is the introduction of a two-phase target features to the contextual based weighting mechanism. The third contribution is the development of a dynamic top-down and bottom-up combination strategy through feature selection. The final contribution is the estimation of a quality score of a novel feature map for dynamic feature map integration.

7.1 Achieved objectives

The following research objectives have been accomplished that contribute towards achieving the overall goal of the thesis:

1. *The first objective was to incorporate high-level knowledge of the scene when modelling top-down saliency to improve the feature weighting process.*
   This has been achieved by proposing a novel approach for modelling top-down saliency that utilized contextual information about an image in learning the attentional feature weights. To our knowledge, this work represents the first of its kind in learning top-down attentional feature weights for target object detection using contextual information. The contextual information represent the gist content of the image and constructed through a feature distribution estimation technique. The model performs a contextual matching between a novel image and a set of learned contextual information through unsupervised clustering. The proposed model referred to as top-down contextual weighting (*TDCoW*) assigns appropriate weights to the features that would maximize the detection accuracy of the target object. The model achieves higher detection accuracy than those that do not utilize contextual information for feature weighting.

2. *The second objective was to improve the detection capabilities of the contextual top-down features weighting approach in more complex and challenging target scenarios.*
   This objective was achieved by proposing a low-level target feature representation combined with contextual weighting. To our knowledge, incorporating low-level target and contextual information to

model top-down attention has not been considered before for target object detection. Two levels of target specific features were introduced, one through target based feature weighting and the other using a target object recognizer. The novelty at both levels comes from their ability to describe and hence detect complex target objects using multi-scale low-level features. Because targets are detected through low-level features, the model is computationally very efficient as no high-level target tuned descriptors or handcrafted features are required. When this two-level target information is combined with the contextual weighting, the detection accuracy is better than a pure contextual based top-down saliency approach.

3. *The third objective was to combine top-down and bottom-up saliencies in a more interactive way to improve the efficiency and detection accuracy over the static combination strategy.*
   This objective was effectively completed by devising an approach that we refer to as feature selection based top-down saliency model (*FS-TDSM*). This approach attempts to combine top-down and bottom-up saliencies in a unique way by formulating the combination procedure as a feature selection problem. Such an approach ensures that the contribution of both saliencies is considered depending on their importance in detecting the target object. Performing a selection from a pool of features belonging to either top-down or bottom-up saliencies results in a more efficient (a smaller number of features) and more accurate (only relevant features) detection of the target. On several challenging datasets, *FS-TDSM* has a better performance than those approaches that combine both saliency features statically.

4. *The fourth objective was to dynamically integrate important feature maps on image basis to avoid the integration of all feature maps for each image.*
   This objective was achieved by proposing a novel model to estimate

a quality score of a feature map through a regression approach. This approach extracts various characteristic features from a feature map that describes its visual attributes and feeds it to a regressor. The learned regression model, in turn, estimates a quality score for the feature map. When this proposed estimation model was applied to a feature integration approach for predicting human fixation, the number of feature maps required was reduced and the human fixation prediction accuracy was boosted. This yielded a dynamic feature map integration mechanism on image basis.

7.2 Conclusions

This section provides important findings, conclusions and limitations of each proposed work in this thesis.

7.2.1 Contextual based top-down saliency weighting

Weighting of top-down attentional features by considering high-level gist information of a scene was introduced for the first time while modelling top-down saliency. The final model achieves better feature weighting than models without context for target object detection.

Each component of this model represents a sub-contribution, which collectively yields the main contribution of this research work. The first sub-contribution was to develop a bottom-up saliency technique based on the *Itti* model of attention. This technique utilizes a new information theoretic centre-surround mechanism called information divergence measure (*IDM*) to yield normalized feature maps. This bottom-up saliency technique was tested on two benchmark datasets for salient object detection. The precision-recall accuracy curves confirm the superiority of our proposed bottom-up saliency technique over several state-of-the-art bottom-up techniques. The reason for proposing a new bottom-up saliency is to have a set of effective features along with a good bottom-up saliency generation mechanism, which is essential in modelling top-down saliency through feature weighting.

The second sub-contribution is the weight computation techniques used in *TDCoW*. We show that more accurate weight values can be produced by computing the weights using Jensen-Shannon divergence *JSD* rather than the conventional *SNR* method. It has been shown the *JSD* is less sensitive to variation in target saliency compared to *SNR* approach. The average precision-recall curve revealed the effectiveness of *JSD* over *SNR* when tested over a dataset of 400 images of cricket balls.

The third sub-contribution is in the form of a contextual clustering framework for assigning feature weights to a novel image. By varying the cluster size for grouping training images with similar context (based on a contextual descriptor matching technique), the detection performance is affected. The average area under the curve (AUC) of the receiver operating characteristic (ROC) shows that as the number of clusters is increased (equivalent to fewer images in a cluster), a better AUC performance is achieved (i.e., better detection accuracy). However, increasing the number of clusters reduces the efficiency of the system because of increase in the number of contextual matchings required for a new image.

TDCoW is the complete solution that combines the above-mentioned components in an effort to use the contextual information for proper weight assignment to the features. The model was evaluated on four datasets, two of cricket balls and the other two from the Graz-02 dataset of persons and bikes. The precision-recall and *F*-measure scores confirms the effectiveness of incorporating contextual information while performing top-down weighting for target object detection. In addition, the model outperforms state-of-the-art bottom-up saliency techniques that model attention in the absence of contextual information.

This model can be generalized to any dataset of images with moderate inter-image content variability. However, if the images in a dataset exhibit high variation in their content (particularly in their background), the learned weights tend to produce uniform weights across all features. This proves to be a potential limitation of this model. Furthermore, the contextual descriptor used in *TDCoW* is large because it is constructed through a kernel density estimator. This reduces the efficiency of the model in the testing phase, particularly when performing descriptor matching with many clusters.

7.2.2 Target information for top-down saliency

Detecting target objects with complex visual attributes is a challenging task. In such scenarios, contextual based attention that provides semantic knowledge of the scene becomes ineffective. More precise knowledge of target characteristics is needed to guide attention towards targets. A model was proposed that combines target information through low-level features with the contextual information in two stages, one through target specific feature weighting and the other through a Naive Bayes recognition model.

The model was tested and analyzed on seven challenging datasets with 12 target objects with varying visual complexity. The results show that the detection accuracy in the form of the F-measure score is always boosted when incorporating target information. When the model uses both stages of target knowledge along with the contextual information, the detection accuracy is maximized.

The proposed model is very efficient as it only utilizes low-level features at multiple scales for describing a target object. These features along with the two stage target feature modelling was able to effectively sample the target objects with high precision. In some occasions, when the target object is very complex (e.g., a multi-part object), the proposed model fails to detect the object. This establishes a limitation on the proposed model as the low-level features provide less description of certain complex target objects.

The proposed approach represents a generalized model for generic target object detection. However, for some complex objects, the detection accuracy decreases due to insufficient features for target description. This could be resolved by introducing more features into the model. Despite this limitation, the recognition module was often able to eliminate false positive regions that were triggered by the model as target objects even when the target was not detected.

7.2.3 Feature selection based saliency combination

An attentional model was proposed that combines bottom-up and top-down processes interactively. The model referred to as feature selection based top-down saliency model (*FS-TDSM*) combines those features belonging to either top-down or bottom-up features by maximizing the *F*-measure score of target detection.

The model is tested on five datasets containing cricket balls as target objects. The detection accuracy is evaluated by averaging the *F*-measure score over all the images in the test set. The performance is also evaluated through the number of fixations required to get to the target object. The proposed combination model is compared with one of the state-of-the-arts visual attention model (referred to as the VOCUS model) that combines top-down and bottom-up processes statically.

The selected features by the proposed model achieve higher detection accuracy when compared to the VOCUS model. The proposed model also requires a smaller number of fixations to reach the target object than those required by the VOCUS model.

A key element in the proposed feature selection approach for combining both saliencies is to put all the features (belonging to either bottom-up or top-down) into a single pool to perform the selection from. This will ensure that the detection of the target is contributed by both saliency processes. This approach also considers those top-down or bottom-up features that are relevant for a detection task. Hence, those features that have a minimal contribution in detecting the target object are removed.

From the obtained selected features, we conclude that both processes contribute actively while performing a visual search for the target object. This contribution is image content dependent. In images with uncluttered backgrounds and few distracting objects, the bottom-up features are preferred. On the other hand, the contribution of these bottom-up features decreases in more complex background images while the importance of top-down features is more prominent. This is in accordance with some of the biological studies that suggest that top-down influence dominates visual search when the target is less discriminant with respect to the background.

The model can be generalized to any target object detection approach provided that appropriate set of top-down target tuned features are considered in the pool of features. Despite the simplicity of the approach, it successfully performed a dynamic combination of top-down and bottom-up saliencies at the features level. We claim that this model is an initial investigation of combining both saliencies dynamically. Hence, the proposed model can be used to build a more complex and dynamic visual attention system for combining top-down and bottom-up saliencies.

7.2.4 Dynamic saliency integration

The main objective of this work was to show whether it is possible to identify the quality of any feature map based on its visual attributes, and how it is possible to perform a dynamic feature map integration for predicting human fixation based on a quality estimation of the feature map.

This objective was successfully achieved by proposing a novel approach for estimating the quality of a feature map (i.e., estimating the AUC-ROC score) through a regression technique. The proposed approach extracts 77 computationally efficient characteristic features from a feature map. These characteristic features describe local and global visual attributes of a feature map through statistical and geometrical information.

A random forest regressor learns a model from these characteristic features to estimate a quality score of a novel feature map accurately without the need of the groundtruth data.

The model was tested on 912 outdoor urban images for predicting human fixation when searching for pedestrians. Particularly, the model was applied to three types of feature maps, a bottom-up map, a target map and a contextual map. The average mean-square-error between the estimated AUC-ROC score and the groundtruth score suggests that our model performs accurate estimation of the AUC-ROC for a given feature map provided that the regression model is trained and tested over a similar type of feature maps.

The model is also flexible in the sense that it extends the estimation to any combination of the above-mentioned feature maps. For any combination of the three feature maps, the model is able to accurately estimate the AUC-ROC score with a high accuracy (i.e., low average MSE), provided that the train/test data type condition mentioned above is not violated.

For each type of feature map, a set of important characteristic features can be inferred by random forest regressor. These characteristic features vary in importance and predictive power from one type of feature map to another. Typically in all three feature maps, it has been found that global statistical characteristic features have the most significance in describing the visual attributes of these feature maps.

The effectiveness of the characteristic features is highly coupled with the structure of these characteristic features in the high dimensional feature space. Some of these complex relations are captured by the feature importance and the partial dependency plots. From this information, it is concluded that the proposed characteristic features, particularly the important ones in each type of feature map, can provide a rough representation of a good feature map for fixation.

The random forest regressor and the proposed set of characteristic features is well suited for the type of feature maps used in this work. The random forest regressor was compared with the kernel based SVM, tree bagging and tree boosting regression techniques. In addition, various previously proposed characteristic features that were used in predicting the quality of feature maps are compared with the proposed set of characteristic features. The average MSE achieved by our model is less than those achieved by other regression techniques and characteristic features, and hence show the effectiveness of our selection for this particular problem.

By integrating our proposed approach with a two simple proposed feature map integration frameworks (CMDT and WMST), we demonstrated that our model achieved a higher human fixation prediction accuracy than that achieved by combining all feature maps. In addition, the proposed integration framework along with the estimation model improves the efficiency of the system as it is not required to compute all three feature maps every time the target object (i.e., pedestrians) is searched for. The model is dynamic as it combines only those feature maps that exhibits high estimated AUC-ROC score on a run time depending on the characteristic features of a feature map.

One limitation of the current estimation model is that it requires the training and testing maps to be similar. instances. This imposes a practical limitation in the CMDT feature map integration framework as it requires a model for each feature map combination. When the number of feature maps increase, it would become impractical to have a regression model trained for each type of feature map combination.

7.3 Future work

Because top-down saliency in this thesis has been modelled for a particular visual task (i.e., target object detection), the selection of target relevant features is critical. In chapter 4, only low-level features were used whereas in the combination model (see chapter 5), handcrafted top-down target specific features were used. For the models to generalize for any complex target object, a mechanism is needed that can generate constructed features from a set of basic low-level and some high-level features to yield more robust, generalized and powerful features automatically. This could be achieved through feature construction and could be considered as a future work.

One limitation with the proposed estimation model is that it has not been tested on scenarios and feature maps other than the case study that comprises of only three feature maps and their combination. Although the characteristic features that are extracted from a feature map represent generalized visual attributes of any given feature map, it is worth investigating the effectiveness of the proposed model on a variety of scenarios and different types of feature maps. It would be interesting to use transfer learning approaches through techniques such as genetic programming [167] and deep neural networks [168] to mitigate the similar instances limitation of the proposed feature map quality estimation model.

Finally, all the proposed works in this thesis aim at modelling topdown saliency without considering the spatial importance of regions of the image. The proposed techniques perform selection over the entire image. These models therefore ignore the fact that different regions of the image might require different attention in terms of what features to extract, what regions to attend to and what features to extract from the next attended region. An active visual attention model is needed that learns the state of the next attended location from the previously attended locations. In this way, efficiency could be further improved by not attending to irrelevant regions of the image and by only computing the features that would best suit the attended location.

Bibliography

[1] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254 – 1259, Nov. 1998.

[2] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145 – 175, May 2001.

[3] K. A. Ehinger, B. Hidalgo-Sotelo, A. Torralba, and A. Olivia, "Modeling search for people in 900 scenes," *Visual Cognition*, vol. 17, no. 6-7, pp. 945 – 978, Jan. 2009.

[4] K. W. Lee and H. Choo, "A critical review of selective attention: an interdisciplinary perspective," *Artificial Intelligence Review*, vol. 40, no. 1, pp. 27 – 50, Jun. 2013.

[5] S. McMains and S. Kastner, "Interactions of top-down and bottom-up mechanisms in human visual cortex," *The Journal of Neuroscience*, vol. 31, no. 2, pp. 587 – 597, Jan. 2011.

[6] D. Gao, S. Han, and N. Vasconcelos, "Discriminant saliency, the detection of suspicious coincidences, and applications to visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 6, pp. 989 – 1005, Jun. 2009.

[7] D. Walther, L. Itti, M. Riesenhuber, T. Poggio, and C. Koch, "Attentional selection for object recognition – a gentle way," in *International Workshop on Biologically Motivated Computer Vision (BMCV)*, vol. 2525, 2002, pp. 472 – 479.

[8] A. Holzbach and G. Cheng, "A fast and scalable system for visual attention, object based attention and object recognition for humanoid robots," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2014, pp. 316 – 321.

[9] G. Wen, B. Rodriguez-Niño, F. Y. Pecen, D. J. Vining, N. Garg, and M. K. Markey, "Comparative study of computational visual attention models on two-dimensional medical images," *Journal of Medical Imaging*, vol. 4, no. 2, p. 025503, May 2017.

[10] M. T. Khanna, K. Rai, S. Chaudhury, and B. Lall, "Perceptual depth preserving saliency based image compression," in *International Conference on Perception and Machine Intelligence (PerMIn)*, 2015, pp. 218 – 223.

[11] T. Cane and J. M. Ferryman, "Saliency-based detection for maritime object tracking," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2016, pp. 1257 – 1264.

[12] Z. Li and L. Itti, "Saliency and gist features for target detection in satellite images," *IEEE Transactions on Image Processing*, vol. 20, no. 7, pp. 2017 – 2029, Jul. 2011.

[13] J. K. Tsotsos *et al.*, "Modeling visual-attention via selective tuning," *Artificial Intelligence*, vol. 78, no. 1-2, pp. 507 – 545, Oct. 1995.

[14] L. Itti and R. J.Peters, "Beyond bottom-up: Incorporating task-dependent influences into a computational model of spatial attention," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007, pp. 1 – 8.

[15] V. Sundstedt, A. Chalmers, K. Debattista, and K. Cater, "Top-down visual attention for efficient rendering of task related scenes," in *International Fall Workshop Vision, Modeling, and Visualization (VMV)*, 2004, pp. 209 – 216.

[16] T. Koike and J. Saiki, "Stochastic guided search model for search asymmetries in visual search tasks," in *International Workshop on Biologically Motivated Computer Vision*, 2002, pp. 408 – 417.

[17] M. Eimer and M. Kiss, "Top-down search strategies determine attentional capture in visual search: Behavioral and electrophysiological evidence," *Attention, Perception, & Psychophysics*, vol. 72, no. 4, pp. 951 – 962, May 2010.

[18] S. Filipe and L. A. Alexandre, "From the human visual system to the computational models of visual attention: a survey," *Artificial Intelligence Review*, Jan. 2013.

[19] S. Frintrop, E. Rome, and H. I. Christensen, "Computational visual attention systems and their cognitive foundations: A survey," *ACM Transactions on Applied Perception*, vol. 7, no. 1, p. Article 6, Jan. 2010.

[20] S. Frintrop, "VOCUS: A visual attention system for object detection and goal-directed search," Ph.D. dissertation, 2006.

[21] S. Mitri, S. Frintrop, K. Pervölz, H. Surmann, and A. Nüchter, "Robust Object Detection at Regions of Interest with an Application in Ball Recognition," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2005, pp. 125 – 130.

[22] V. Navalpakkam and L. Itti, "An integrated model of top-down and bottom-up attention for optimizing detection speed," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006, pp. 2049 – 2056.

[23] B. Rasolzadeh, A. T. Targhi, and J.-O. Eklundh, "An attentional system combining top-down and bottom-up influences," in *International Workshop on Attention in Cognitive Systems (WAPCV)*, 2007, pp. 123 – 140.

[24] A. X. Benicasa, M. G. Quiles, L. Zhao, and R. A. F. Romero, "Top-down biasing and modulation for object-based visual attention," in *International conference on neural information processing (ICONIP)*, 2013, pp. 325 – 332.

[25] J. Yang and M.-H. Yang, "Top-Down visual saliency via joint CRF and dictionary learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 3, pp. 576 – 588, Mar. 2017.

[26] S. He and R. W. H. Lau, "Exemplar-driven top-Down saliency detection via deep association," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* , 2016, pp. 5723 – 5732.

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

[28] M. Cerf, J. Harel, W. Einhaeuser, and C. Koch, "Predicting human gaze using low-level saliency combined with face detection," in *Advances in Neural Information Processing Systems (NIPS)*, 2007, pp. 241 – 248.

[29] L. Itti, N. Dhavale, and F. Pighin, "Realistic avatar eye and head animation using a neurobiological model of visual attention," in *SPIE Annual International Symposium on Optical Science and Technology*, 2003, pp. 64 – 78.

[30] S. Spotorno, G. L. Malcolm, and B. W. Tatler, "How context information and target information guide the eyes from the first epoch

of search in real world scenes," *Journal of Vision*, vol. 14, no. 2, p. (Article 7), 2014.

[31] U. Rutishauser, D. Walther, C. Koch, and P. Perona, "Is bottom-up attention useful for object recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2004, pp. 37 – 44.

[32] J. Kim, H. Lee, and J. Kim, "A novel method for salient object detection via compactness measurement," in *IEEE International Conference on Image Processing (ICIP)*, 2013, pp. 3426 – 3430.

[33] H. Li, H. Lu, Z. Lin, X. Shen, and B. Price, "Inner and inter label propagation: Salient object detection in the wild," *IEEE Transactions on Image Processing*, vol. 24, no. 10, pp. 3176 – 3186, Jun. 2015.

[34] S. Chen, Y. Li, and N. M. Kwok, "Active vision in robotic systems: A survey of recent developments," *International Journal of Robotics Research*, vol. 30, no. 11, pp. 1343 – 1377, Sep. 2011.

[35] A. Andreopoulos and J. K. Tsotsos, "A computational learning theory of active object recognition under uncertainty," *International Journal of Computer Vision*, vol. 101, no. 1, pp. 95 – 142, Jan. 2013.

[36] D. Ognibene and G. Baldassare, "Ecological active vision: Four bioinspired principles to integrate bottom–Up and adaptive top–Down attention tested with a simple camera-arm robot," *IEEE Transactions on Autonomous Mental Development*, vol. 7, no. 1, pp. 3 – 25, Mar. 2015.

[37] A. L. Rothenstein and J. Tsotsos, "Attention links sensing to recognition," *Image and Vision Computing*, vol. 26, no. 1, pp. 114 – 126, Jan. 2008.

[38] J. K. Tsotsos and A. Rothenstein, "Computational models of visual attention," *Scholarpedia*, vol. 6, no. 1, p. 6201, 2011.

[39] A. Anzai, X. Peng, and D. C. Van Essen, "Neurons in monkey visual area V2 encode combinations of orientations," *Nat Neurosci*, vol. 10, no. 10, pp. 1313 – 1321, Oct. 2007.

[40] J. Theeuwes, "Top-down search strategies cannot override attentional capture," *Psychonomic Bulletin & Review*, vol. 11, no. 1, pp. 65 – 70, Feb. 2004.

[41] A. Treisman and G. Gelade, "A feature integration theory of attention," *Cognitive Psychology*, vol. 12, no. 1, pp. 97 – 136, 1980.

[42] C. Koch and S. Ullman, "Shifts in selective visual attention: Towards the underlying neural circuitry," *Hum. Neurobiology*, vol. 4, no. 4, pp. 219 – 227, 1985.

[43] O. Le Meur, P. Le Callet, D. Barba, and D. Thoreau, "A coherent computational approach to model bottom-up visual attention," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 5, pp. 802 – 817, 2006.

[44] O. Le Meur, P. Le Callet, and D. Barba, "Predicting visual fixations on video based on low-level visual features," *Vision Research*, vol. 47, no. 19, pp. 2483 – 2498, Sep. 2007.

[45] P. Bian and L. Zhang, "Biological plausibility of spectral domain approach for spatiotemporal visual saliency," in *International Conference on Advances in Neuro-information Processing (ICONIP)*, vol. 5506, 2009, pp. 251 – 258.

[46] S. He, J. Han, X. Hu, M. Xu, L. Guo, and T. Liu, "A biologically inspired computational model for image saliency detection," in *ACM International Conference on Multimedia*, 2011, pp. 1465 – 1468.

[47] I. Rigas, G. Economou, and S. Fotopoulos, "Low-level visual saliency with application on aerial imagery," *IEEE Geoscience and Remote Sensing Letters*, vol. 10, no. 6, pp. 1389 – 1393, Nov. 2013.

[48] N. Murray, M. Vanrell, X. Otazu, and C. A. Parraga, "Saliency estimation using a non-parametric low-level vision model," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 433 – 440.

[49] A. Andreopoulos and J. Tsotsos, "50 Years of object recognition: Directions forward," *Computer Vision and Image Understanding*, vol. 117, no. 8, pp. 827 – 891, Aug. 2013.

[50] A. Bogdan, D. Thomas, and F. Vittorio, "Measuring the objectness of image windows," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2189 – 2202, Nov. 2012.

[51] "Caltech101." [Online]. Available: http://www.vision.caltech.edu/Image_Datasets/Caltech101/Caltech101.html

[52] "PASCAL VOC." [Online]. Available: http://pascallin.ecs.soton.ac.uk/challenges/VOC/

[53] A. Królak and P. Strumillo, "Eye-blink detection system for human-computer interaction," *Universal Access in the Information Society*, vol. 11, no. 4, pp. 409 – 419, Nov. 2012.

[54] Y. Wu, J. Lim, and M.-H. Yang, "Object tracking benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1834 – 1848, Sep. 2015.

[55] J. Li, N. M. Allinson, D. Tao, and X. Li, "Multitraining support vector machine for image retrieval," *IEEE Transactions on Image Processing*, vol. 15, no. 11, pp. 3597 – 3601, Nov. 2006.

[56] Y.-C. Chung and Z. He, "Low-complexity and reliable moving objects detection and tracking for aerial video surveillance with small UAVS," in *IEEE International Symposium on Circuits and Systems (ISCAS)* , 2007, pp. 2670 – 2673.

[57] I. Ulusoy and C. M. Bishop, *Comparison of Generative and Discriminative Techniques for Object Detection and Classification*, ser. Lecture Notes in Computer Science. Springer-Verlag, 2006, vol. 4170, pp. 173 – 195.

[58] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627 – 1645, Sep. 2010.

[59] H. Azizpour and I. Laptev, "Object detection using strongly-supervised deformable part models," in *European Conference on Computer Vision (ECCV)*, Oct. 2012, pp. 836 – 849.

[60] J.-M. Kim and M. A. Kang, "Appearance-based object recognition using higher correlation feature information and PCA," in *IEEE International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, 2011, pp. 1874 – 1878.

[61] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91 – 110, Nov. 2004.

[62] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2005, pp. 886 – 893.

[63] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "SURF: Speeded up robust features," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346 – 359, Jun. 2008.

[64] C. Harris and M. Stephens, "A combined corner and edge detector," in *Fourth Alvey Vision Conference*, 1988, pp. 147 – 151.

[65] J. Sivic and A. Zisserman, "Video google: a text retrieval approach to object matching in videos," in *IEEE International Conference on Computer Vision (ICCV)*, vol. 24, Oct. 2003, pp. 1470 – 1477.

[66] A. Torralba, "Contextual priming for object detection," *International Journal of Computer Vision*, vol. 53, no. 2, pp. 169 – 191, Jul. 2003.

[67] A. Torralba, K. Murphy, W. Freeman, and M. Rubin, "Context-based vision system for place and object recognition," in *IEEE International Conference on Computer Vision (ICCV)*, 2003, pp. 273 – 280.

[68] L. Wolf and S. Bileschi, "A critical view of context," *International Journal of Computer Vision*, vol. 69, no. 2, pp. 251 – 261, Aug. 2006.

[69] P. Viola and M. Jones, "Robust Real-Time Face Detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137 – 154, May 2004.

[70] C. Siagian and L. Itti, "Rapid biologically-inspired scene classification using features shared with visual attention," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 300 – 312, Feb. 2007.

[71] A. Borji and L. Itti, "Scene classification with a sparse set of salient regions," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 1902 – 1908.

[72] A. Lavelli, M. E. Califf, F. Ciravegna, D. Freitag, C. Giuliano, N. Kushmerick, L. Romano, and N. Ireson, "Evaluation of machine learning-based information extraction algorithms: criticisms and recommendations," *Language Resources and Evaluation*, vol. 42, no. 4, pp. 361 – 393, Dec. 2008.

[73] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look," in *IEEE International Conference on Computer Vision (ICCV)*, 2009, pp. 2106 – 2113.

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

[75] N. D. B. Bruce and J. K. Tsotsos, "Saliency based on information maximization," in *Advances in Neural Information Processing Systems (NIPS)*, 2006, pp. 155 – 162.

[76] G. Stas, Z.-M. Lihi, and T. Ayellet, "Context-aware saliency detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 10, pp. 1915 – 1926, Oct. 2012.

[77] O. Le Meur and T. Baccino, "Methods for comparing scanpaths and saliency maps: strengths and weaknesses," *Behavior Research Methods*, vol. 45, no. 1, pp. 251 – 266, Mar. 2013.

[78] T. Judd, F. Durand, and A. Torralba, "A benchmark of computational models of saliency to predict human fixations," Tech. Rep., Jan. 2012.

[79] T. Liu, J. Sun, N.-N. Zheng, X. Tang, and H.-Y. Shum, "Learning to detect a salient object," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 2, pp. 353 – 367, Feb. 2011.

[80] J. Li, M. D. Levine, X. An, X. Xu, and H. He, "Visual saliency based on scale-space analysis in the frequency domain," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 4, pp. 996 – 1010, Apr. 2013.

[81] J. Wang, D. M. Chandler, and P. L. Callet, "Quantifying the relationship between visual salience and visual importance," *SPIE Human and Electronic Imaging (HVEI) XV*, vol. 7527, pp. 1 – 30, Aug. 2010.

[82] V. Movahedi and J. H. Elder, "Design and perceptual validation of performance measures for salient object segmentation," in *IEEE*

Computer Society Workshop on Perceptual Organization in Computer Vision (POCV), 2010, pp. 49 – 56.

[83] S. Alpert, M. Galun, A. Brandt, and R. Basri, "Image segmentation by probabilistic bottom-up aggregation and cue integration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 2, pp. 315 – 327, Feb. 2012.

[84] H. Greenspan, S. Belongie, R. M. Goodman, P. Perona, S. Rakshit, and C. H. Anderson, "Overcomplete steerable pyramid filters and rotation invariance." in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1994, pp. 222 – 228.

[85] G. Kootstra, B. De Boer, and L. R. B. Schomaker, "Predicting eye fixations on complex visual stimuli using local symmetry," *Cognitive Computation*, vol. 3, no. 1, pp. 223 – 240, Mar. 2011.

[86] R. Achanta and S. Susstrunk, "Saliency detection using maximum symmetric surround," in *IEEE International Conference on Image Processing (ICIP)*, 2010, pp. 2653 – 2656.

[87] X. Hou and L. Zhang, "Saliency detection: A spectral residual approach," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007, pp. 1 – 8.

[88] J. Xu, "Bayesian modeling of visual attention," in *International Conference on Advances in Neural Information Processing (ICONIP)*, vol. 7664, 2012, pp. 92 – 99.

[89] L. Itti and P. Baldi, "Bayesian surprise attracts human attention," *Vision Research*, vol. 49, no. 10, pp. 1295 – 306, Jun. 2009.

[90] G. Wang and D. A. Forsyth, "Joint learning of visual attributes, object classes and visual saliency," in *IEEE International Conference on Computer Vision (ICCV)*, 2009, pp. 537 – 544.

[91] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," in *Advances in Neural Information Processing Systems (NIPS)*, 2006, pp. 545 – 552.

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

[93] J. Zhou and Z. Jin, "A new framework for multiscale saliency detection based on image patches," *Neural Processing Letters*, vol. 38, no. 3, pp. 361 – 374, Dec. 2013.

[94] L. Duan, C. Wu, J. Miao, L. Qing, and Y. Fu, "Visual saliency detection by spatially weighted dissimilarity," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 473 – 480.

[95] W. Wang, Q. Huang, and W. Gao, "Mesaring visual saliency by site entropy," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 2368 – 2375.

[96] W. Houa, X. Gaoa, D. Taob, and X. Li, "Visual saliency detection using information divergence," *Pattern Recognition*, vol. 46, no. 10, pp. 2658 – 2669, Oct. 2013.

[97] A. Borji and L. Itti, "State-of-the-art in visual attention modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 185 – 207, Jan. 2013.

[98] J. M. Wolfe, T. S. Horowitz, N. Kenner, M. Hyle, and N. Vasan, "How fast can you change your mind? the speed of top-down," *Vision Research*, vol. 44, no. 12, pp. 1411 – 1426, Jun. 2004.

[99] T. J. Vickery, L.-W. King, and Y. Jiang, "Setting up the target template in visual search," *Journal of Vision*, vol. 5, no. 1, pp. 81 – 92, Feb. 2005.

[100] L. Zhaoping and U. Frith, "A clash of bottom-up and top-down processes in visual search: the reversed letter effect revisited," *Journal of Experimental Psychology, Human Perception and Performan*, vol. 37, no. 4, pp. 997 – 1006, Aug. 2011.

[101] M. S. Castelhano, M. L. Mack, and J. M. Henderson, "Viewing task influences eye movement control during active scene perception," *Journal of Vision*, vol. 9, no. 3, pp. 1 – 15 (Article 6), Mar. 2009.

[102] G. L. Malcolm and J. M. Henderson, "Combining top-down processes to guide eye movements during real-world scene search," *Journal of Vision*, vol. 10, no. 2, pp. 1 – 11 (Article 4), 2010.

[103] W. Einhaeuser, U. Rutishauser, and C. Koch, "Task-demands can immediately reverse the effects of sensory-driven saliency in complex visual stimuli," *Journal of Vision*, vol. 8, no. 2, pp. 1 – 19 (Article 2), Feb. 2008.

[104] A. Nuthmann and J. M. Henderson, "Object-based attentional selection in scene viewing," *Journal of Vision*, vol. 10, no. 8, p. (Article 20), Jul. 2010.

[105] W. Einhaeuser, M. Spain, and P. Perona, "Objects predict fixations better than early saliency," *Journal of Vision*, vol. 8, no. 14, p. (Article 18), 2008.

[106] S. Frintrop, G. Backer, and E. Rome, "Goal-directed search with a top-down modulated computational attention system," in *Joint Pattern Recognition Symposium DAGM: Pattern Recognition*, ser. Lecture Notes in Computer Science, vol. 3663, 2005, pp. 117 – 124.

[107] S. Frintrop, P. Jensfelt, and H. I. Christensen, "Pay attention when selecting features," in *International Conference on Pattern Recognition (ICPR)*, vol. 2, 2006, pp. 163 – 166.

[108] V. Navalpakkam and L. Itti, "Optimal cue selection strategy," in *Advances in Neural Information Processing Systems (NIPS)*, 2005, pp. 987 – 994.

[109] Y. Hu, X. Xie, W.-Y. Ma, L.-T. Chia, and D. Rajan, "Salient region detection using weighted feature maps based on the human visual attention model," in *Advances in Multimedia Information Processing (PCM)*, 2004, pp. 993 – 1000.

[110] A. J. Palomino, R. Marfil, J. P. Bandera, and A. Bandera, "Multi-feature bottom-up processing and top-down selection for an object-based visual attention model," in *Workshop on Recognition and Action for Scene Understanding (REACTS)*, Aug. 2013, pp. 29 – 43.

[111] B. Rasolzadeh, M. Björkman, K. Huebner, and D. Kragic, "An active vision system for detecting, fixating and manipulating objects in the real world," *International Journal of Robotics Research*, vol. 29, no. 2 – 3, pp. 133 – 154, Feb. 2010.

[112] G. Zhua, Q. Wanga, and Y. Yuan, "Tag-saliency: Combining bottom-up and top-down information for saliency detection," *Computer Vision and Image Understanding*, vol. 118, pp. 40 – 49, Jan. 2014.

[113] L. Zhang, M. H. Tong, T. K. Marks, H. Shan, and G. W. Cottrell, "SUN: A Bayesian framework for saliency using natural statistics," *Journal of Vision*, vol. 8, no. 7, pp. 1 – 20 (Article 32), Dec. 2008.

[114] S.-b. Choi, S.-w. Ban, and M. Lee, "Biologically motivated visual attention system using bottom-up saliency map and top-down inhibition," *Neural Information Processing-Letters and Reviews*, vol. 2, no. 1, pp. 19 – 25, Jan. 2004.

[115] M. Fornoni and B. Caputo, "Indoor scene recognition using task and saliency-driven feature pooling," in *British Machine Vision Conference (BMVC)*, Sep. 2012.

[116] Q. Zhao and C. Koch, "Learning saliency-based visual attention: A review," *Signal Processing*, vol. 93, no. 6, pp. 1401 – 1407, Jun. 2013.

[117] D. Gao and N. Vasconcelos, "Integrated learning of saliency, complex features and object detectors from cluttered scenes," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005, pp. 282 – 287.

[118] C. A. Rothkopf, D. H. Ballard, and M. M. Hayhoe, "Task and context determine where you look," *Journal of Vision*, vol. 7, no. 14, pp. 1 – 20 (Article 16), Dec. 2007.

[119] A. Torralba, A. Olivia, M. S. Castelhano, and J. M. Henderson, "Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search," *Psychological Review*, vol. 113, no. 4, pp. 766 – 786, Oct. 2006.

[120] I. M. H. Rahman, C. Hollitt, and M. Zhang, "Information divergence based saliency detection with a global center-surround mechanism," in *International Conference on Pattern Recognition (ICPR)*, 2014, pp. 3428 – 3433.

[121] L. Itti and C. Koch, "Feature combination strategies for saliency-based visual attention system," *Journal of Electronic Imaging*, vol. 10, no. 1, pp. 161 – 169, Jan. 2001.

[122] T. Li, T. Mei, I.-S. Kweon, and X.-S. Hua, "Contextual bag-of-words for visual categorization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 4, pp. 381 – 392, Apr. 2011.

[123] L. Zhou, Z. Zhou, and D. Hu, "Scene classification using a multi-resolution bag-of-features model," *Pattern Recognition*, vol. 46, no. 1, pp. 424 – 433, Jan. 2013.

[124] A. Borji, D. N. Sihite, and L. Itti, "Salient object detection: A benchmark," in *European Conference on Computer Vision (ECCV)*, 2012, pp. 414 – 429.

[125] N. Riche, M. Mancas, M. Duvinage, M. Mibulumukin, B. Gosselin, and T. Dutoit, "RARE2012: A multi-scale rarity-based saliency detection with its com parative statistical analysis," *Signal Processing: Image Communication*, vol. 28, no. 6, pp. 3114 – 3124, Mar. 2013.

[126] N. Tong, H. Lu, X. Ruan, and M.-H. Yang, "Salient object detection via bootstrap learning," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 1884 – 1892.

[127] Y. Qin, H. Lu, Y. Xu, and H. Wang, "Saliency detection via cellular automata," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 110 – 119.

[128] C. Li, Y. Yuan, W. Cai, Y. Xia, and D. Feng, "Robust saliency detection via regularized random walks ranking," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 2710 – 2717.

[129] P. Jiang, H. Ling, J. Yu, and J. Peng, "Salient region detection by UFO: uniqueness, focusness and objectness," in *IEEE International Conference on Computer Vision (ICCV)*, 2013, pp. 1976 – 1983.

[130] M.-M. Cheng, J. Warrell, W.-Y. Lin, S. Zheng, V. Vineet, and N. Crook, "Efficient salient region detection with soft image abstraction," in *IEEE International Conference on Computer Vision (ICCV)*, 2013, pp. 1529 – 1536.

[131] C. Aytekin, S. Kiranyaz, and M. Gabbouj, "Automatic object segmentation by quantum cuts," in *International Conference on Pattern Recognition (ICPR)*, 2014, pp. 112 – 117.

[132] J. Kim, D. Han, Y.-W. Tai, and J. Kim, "Salient region detection via high-dimensional color transform," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2014, pp. 883 – 890.

[133] C. Yang, L. Zhang, H. Lu, and M. Yang, "Saliency detection via graph-based manifold ranking," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 3166 – 3173.

[134] K. Gu, S. J. Tong, G. Zhai, W. Lin, X. Yang, and W. Zhang, "Visual saliency detection with free energy theory," *IEEE signal processing letters*, vol. 22, no. 10, pp. 1552 – 1555, Mar. 2015.

[135] A. Opelt, A. Pinz, M. Fussenegger, and P. Auer, "Generic object recognition with boosting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 3, pp. 416 – 431, Mar. 2006.

[136] J. Mutch and D. G. Lowe, "Multiclass object recognition with sparse, localized features," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006, pp. 11 – 18.

[137] J. Hu, Y. Wang, E. Zhou, M. C. Fu, and S. I. Marcus, *A survey of some model-based methods for global optimization*. Birkhäuser Boston, 2012, pp. 157 – 179.

[138] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *IEEE International Conference on Neural Networks*, vol. 4, 1995, pp. 1942 – 1948.

[139] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimization for feature selection in classification: A multi-objective approach," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1656 – 1671, Dec. 2013.

[140] M. Zambrano-Bigiarini, M. Clerc, and R. Rojas, "Standard particle swarm optimisation 2011 at CEC-2013: A baseline for future PSO

improvements," in *IEEE Congress on Evolutionary Computation (CEC)*, 2013, pp. 2337 – 2344.

[141] A. P. Engelbrecht, "Particle swarm optimization: Global best or local best?" in *Brazilian Congress on Computational Intelligence (BRICS-CCI & CBIC)*, 2013, pp. 124 – 135.

[142] S. Cheng, Y. Shi, and Q. Qin, "Population diversity of particle swarm optimizer solving single and multi-objective problems," *International Journal of Swarm Intelligence Research (IJSIR)*, vol. 3, no. 4, pp. 23 – 60, 2012.

[143] V. Navalpakkam and L. Itti, "Modeling the influence of task on attention," *Vision Research*, vol. 45, no. 2, pp. 205 – 231, Jan. 2005.

[144] N. Pinto, D. D. Cox, and J. J. DiCarlo, "Why is real-world visual object recognition hard," *PLoS Computational Biology*, vol. 4, no. 1, p. e27, Jan. 2008.

[145] R. C. González and R. E. Woods, *Digital image processing*, 3rd ed. Pearson Education, 2008.

[146] G. Loy and A. Zelinsky, "Fast radial symmetry for detecting points of interest," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp. 959 – 973, Aug. 2003.

[147] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898 – 916, May 2011.

[148] L. Cervante, B. Xue, M. Zhang, and L. Shang, "Binary particle swarm optimisation for feature selection: A filter based approach," in *IEEE Congress on Evolutionary Computation (CEC)*, 2012, pp. 1 – 8.

[149] J. Kennedy and R. Eberhart, “A discrete binary version of the particle swarm algorithm,” in *IEEE International Conference on Systems, Man, and Cybernetics*, vol. 5, 1997, pp. 4104 – 4108.

[150] L. Cervante, B. Xue, L. Shang, and M. Zhang, “A multi-objective feature selection approach based on binary PSO and rough set theory,” in *European conference on Evolutionary Computation in Combinatorial Optimization (EvoCOP)*, vol. 7832, 2013, pp. 25 – 36.

[151] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, no. 1, pp. 62 – 66, Jan. 1979.

[152] V. Gopalakrishnan, Y. Hu, and D. Rajan, “Salient region detection by modeling distributions of color and orientation,” *IEEE Transactions on Multimedia*, vol. 11, no. 5, pp. 892 – 905, Aug. 2009.

[153] Q. Yan, L. Xu, J. Shi, and J. Jia, “Hierarchical saliency detection,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 1155 – 1162.

[154] R. L. Canosa, “Real-world vision: Selective perception and task,” *ACM Transactions on Applied Perception*, vol. 6, no. 2, p. Article 11, Feb. 2009.

[155] S. Karthikeyan, V. Jagadeesh, and B. S. Manjunath, “Learning top down scene context for visual attention modeling in natural images,” in *IEEE International Conference on Image Processing (ICIP)*, 2013, pp. 211 – 215.

[156] S. Ren, X. Cao, Y. Wei, and J. Sun, “Face alignment via regressing local binary features,” *IEEE Transactions on Image Processing*, vol. 25, no. 3, pp. 1233 – 1245, Mar. 2016.

[157] I. Naseem, R. Togneri, and M. Bennamoun, "Linear regression for face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 11, pp. 2106 – 2112, Nov. 2010.

[158] G. Fanelli, M. Dantone, J. Gall, A. Fossati, and L. J. V. Gool, "Random forests for real time 3D face analysis," *International Journal of Computer Vision*, vol. 101, no. 3, pp. 437 – 458, Feb. 2013.

[159] Y. Amit and D. Geman, "Shape quantization and recognition with randomized trees," *Neural Computation*, vol. 9, no. 7, pp. 1545 – 1588, Oct. 1997.

[160] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 5, pp. 5 – 32, Oct. 2001.

[161] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and regression trees*. Chapman & Hall, 1984.

[162] G. Biau, "Analysis of a random forests model," *Journal of Machine Learning Research*, vol. 13, no. 1, pp. 1063 – 1095, Jan. 2012.

[163] L. J. P. van der Maaten and G. E. Hinton, "Visualizing high-dimensional data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579 – 2605, Nov. 2008.

[164] http://cvcl.mit.edu/searchmodels/.

[165] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123 – 140, Aug. 1996.

[166] T. R. T. Hastie and J. Friedman, *The elements of statistical learning*, 2nd ed. Springer, 2008, ch. Model assessment and selection.

[167] M. Iqbal, B. Xue, H. Al-Sahaf, and M. Zhang, "Cross-domain reuse of extracted knowledge in genetic programming for image classification," *IEEE Transactions on Evolutionary Computation*, vol. xx, no. x, pp. xxx – xxx, 2017.

[168] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345 – 1359, Oct. 2010.