Advanced Visual Computing for Image Saliency Detection

THE UNIVERSITY OF SYDNEY

Yuchen Yuan
SID 440409539

Supervisor: Prof. David Dagan Feng
A/Supervisor: A/Prof. Weidong (Tom) Cai

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

School of Information Technologies
Faculty of Engineering & Information Technologies
The University of Sydney

March 2017

Abstract

Saliency detection is a category of computer vision algorithms that aims to filter out the most salient object in a given image. Existing saliency detection methods can generally be categorized as bottom-up methods and top-down methods, and the prevalent deep neural network (DNN) has begun to show its applications in saliency detection in recent years. However, the challenges in existing methods, such as problematic pre-assumption, inefficient feature integration and absence of high-level feature learning, prevent them from superior performances.

In this thesis, to address the limitations above, we have proposed multiple novel models with favorable performances. Specifically, we first systematically reviewed the developments of saliency detection and its related works, and then proposed four new methods, with two based on low-level image features, and two based on DNNs. The regularized random walks ranking method (RR) and its reversion-correction-improved version (RCRR) are based on conventional low-level image features, which exhibit higher accuracy and robustness in extracting the image boundary based foreground / background queries; while the background search and foreground estimation (BSFE) and dense and sparse labeling (DSL) methods are based on DNNs, which have shown their dominant advantages in high-level image feature extraction, as well as the combined strength of multi-dimensional features. Each of the proposed methods is evaluated by extensive experiments, and all of them behave favorably against the state-of-the-art, especially the DSL method, which achieves remarkably higher performance against sixteen state-of-the-art methods (including ten conventional methods and six

learning based methods) on six well-recognized public datasets. The successes of our proposed methods reveal more potential and meaningful applications of saliency detection in real-life computer vision tasks.

Declaration

I hereby declare that to the best of my knowledge, the content of this thesis is my own work. This thesis has not been submitted for any other degree or academic award. I certify that the intellectual content of this thesis is the product of my own work, and that all the assistances I received in preparing this thesis have been acknowledged.

For the content of this thesis, Chapter 3 is published on [1]; I designed the main part of the algorithm, conducted all of the experiments, and wrote the draft. Chapter 4 is published on [2]; I designed the entire algorithm, conducted all of the experiments, and wrote the draft. Chapter 5.4 and 5.5 are published on [3]; I designed part of the algorithm, and conducted part of the experiments. Chapter 5.6 and 5.7 are published on [4]; I designed the entire algorithm, conducted all of the experiments, and wrote the draft.

In addition to the statements above, in cases where I am not the corresponding author of a published item, permission to include the published material has been granted by the corresponding author.

Student Name:	Date:
Yuchen Yuan	31.03.2017

Acknowledgements

I would like to express my sincere gratitude to all the individuals and organizations that have provided invaluable help to my PhD study.

First and foremost, I would like to show the deepest appreciation to my primary supervisor, Prof. David Dagan Feng, for his all-around guidance and support throughout my entire PhD study. His broad knowledge and keen perspective about the future research trends have significantly inspired me for pursuing the cutting edge technologies in my research field, as well as making self-actualization as the ultimate goal for my life. It is my great honor to be his student.

I am particularly grateful to my associate supervisor, A/Prof. Weidong (Tom) Cai, for his comprehensive guidance to my research works. It is because of him that I have acquired considerable critical skills of detail handling in scientific researches. More importantly, his advisements reach beyond research itself, and lead me to become a better person. I consider myself very fortunate to have the opportunity working with him.

My appreciation extends to A/Prof. Jinman Kim for his generous assistance in my paper revisions, research topic discussions, and especially his recommendation of me to the internship at Microsoft Research Asia (MSRA).

I am grateful to Dr. Changyang Li for his instructions and supports to my research. Thanks to him, I was able to quickly gather the necessary knowledge at the beginning of my PhD study. And because of the high performance computing machine he provided, my research works of deep neural networks have been notably facilitated.

I would like to thank my cooperators at Shanghai Jiaotong University, including Prof. Zeguang Han, Dr. Yi Shi, A/Prof. Xin Zou and Dr. Xianbin Su, for their great help of my involvements in bioinformatics and genomics. I am looking forward to the bright prospects of the USYD-SJTU Research Alliance.

I greatly appreciate the supervision of A/Dean Eric Chang and A/Prof. Yan Xu during my internship at MSRA. They have provided me a highly intensive industry training, which not only promoted my technical skills immensely, but also readies me for the challenges in my future works.

I would also like to thank my fellow students at the Biomedical and Multimedia Information Technology (BMIT) group, especially Ke Yan and Lei Bi, for their help during my daily researches.

Furthermore, I gratefully acknowledge the funding sources that have made my PhD study possible, including University of Sydney International Scholarship (USydIS), University of Sydney and Shanghai Jiaotong University (USYD-SJTU) Joint Research Alliance, and Australian Research Council (ARC).

Last but not least, I would like to show my gratitude to my family, including my grandparents, my parents, and my little sister, for their continuous support throughout my whole life. I am grateful to my girlfriend Alice for her unconditional trust and support, especially during the most difficult period of my study.

List of Publications

Journal Papers:

1. **Y. Yuan**, C. Li, J. Kim, W. Cai, and D. D. Feng. "Reversion correction and regularized random walks ranking for saliency detection," *IEEE Transactions on Image Processing.* (accepted to appear)
2. **Y. Yuan**, C. Li, J. Kim, W. Cai, and D. D. Feng, "Dense and sparse labeling with multi-dimensional features for saliency detection," *IEEE Transactions on Circuits and Systems for Video Technology*. (accepted to appear)
3. **Y. Yuan**, Y. Shi, C. Li, J. Kim, W. Cai, Z. Han, *et al.*, "DeepGene: an advanced cancer type classifier based on deep learning and somatic point mutations," *BMC Bioinformatics*, vol. 17, no. 17, pp. 243-256, 2016.
4. **Y. Yuan**, C. Li, and D. D. Feng. "Energy-driven smoothing and prior statistical approximation for image segmentation," *IEEE Transactions on Circuits and Systems for Video Technology*. (under major revision)

Conference Papers:

5. C. Li, **Y. Yuan**, W. Cai, Y. Xia, and D. D. Feng, "Robust saliency detection via regularized random walks ranking," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 2710-2717.

6. **Y. Yuan**, Y. Shi, C. Li, J. Kim, W. Cai, *et al.*, "DeepGene: an advanced cancer type classifier based on deep learning and somatic point mutations," in *International Conference on Genome Informatics (GIW)*, Shanghai, China, Oct. 2016, no. 96.
7. **Y. Yuan**, Y. Shi, X. Su, X. Zou, Q. Luo, *et al*., "Copy number aberration based cancer type prediction with convolutional neural networks," in *International Symposium on Bioinformatics Research and Applications (ISBRA)*, Honolulu, HI, USA, May. 2017, track 2, no. 10.
8. K. Yan, C. Li, X. Wang, **Y. Yuan**, A. Li, *et al.*, "Comprehensive autoencoder for prostate recognition on MR images," in *IEEE International Symposium on Biomedical Imaging (ISBI)*, Prague, Czech Republic, Apr. 2016, pp. 1190-1194.
9. K. Yan, C. Li, X. Wang, A. Li, **Y. Yuan**, *et al*., "Adaptive background search and foreground estimation for saliency detection via comprehensive autoencoder," in *IEEE International Conference on Image Processing (ICIP)*, Phoenix, AZ, USA, Sep. 2016, pp. 2767-2771.
10. K. Yan, C. Li, X. Wang, A. Li, **Y. Yuan**, *et al*., "Automatic prostate segmentation on MR images with deep network and graph model," in *IEEE Internaltional Conference of Engineering in Medicine and Biology Society (EMBC)*, Orlando, FL, USA, Aug. 2016, pp. 635-638.

Contents

List of Figures 14

List of Tables 20

1 Introduction 22

1.1 Background of Saliency Detection 22

1.2 Significance of Research 24

1.3 Existing Challenges 25

1.3.1 Problematic Pre-assumptions 25

1.3.2 Ineffective Feature Integration 26

1.3.3 Absence of High-Level Feature Abstraction and Learning 27

1.4 Contributions 28

1.4.1 Conventional Low-Level Feature Based Saliency Detection 29

1.4.2 Improved Low-Level Feature Based Saliency Detection 29

1.4.3 Deep Neural Network Based Saliency Detection 30

1.5 Thesis Organization 32

2 Related Works 34

2.1 Saliency Detection 34

2.1.1 Bottom-Up Methods 34

2.1.2 Top-Down Methods 38

2.1.3 Other Methods 40

2.2 Image Segmentation 41

2.3 Object Proposal Generation 45

2.4 Deep Neural Network (DNN) 46

2.4.1 Fundamentals of Neural Networks 46
2.4.2 DNN Based Sparse Labeling 50
2.4.3 DNN Based Dense Labeling 51
3 Conventional Low-Level Feature Based Saliency Detection 52
3.1 Problem Formulation 52
3.2 Contributions 53
3.3 Related Works 53
3.3.1 Manifold Ranking 54
3.3.2 Random Walks 56
3.4 Saliency Detection with Regularized Random Walks Ranking (RR) 57
3.4.1 Background Saliency Estimation 58
3.4.2 Foreground Saliency Estimation 60
3.4.3 Saliency Map Formulation by Regularized Random Walks Ranking 60
3.5 Experimental Results 63
3.5.1 Datasets 63
3.5.2 Evaluation Metrics 64
3.5.3 Parameters 65
3.5.4 Implementation 65
3.5.5 Evaluation of Design Options 65
3.5.6 Evaluation Against State-of-the-Art 66
3.5.7 Efficiency 74
3.5.8 Limitation 74
3.6 Summary 74
4 Improved Low-Level Feature Based Saliency Detection 76
4.1 Problem Formulation 76

4.2 Contributions 78
4.3 Related Works 79
4.3.1 *K*-Means Clustering 80
4.4 Saliency Detection with Reversion Correction and Regularized Random Walks Ranking (RCRR) 80
4.4.1 Saliency Reversion Correction 81
4.4.2 Regularized Random Walks Ranking 84
4.5 Experimental Results 88
4.5.1 Datasets 88
4.5.2 Evaluation Metrics 89
4.5.3 Parameters 91
4.5.4 Implementation 92
4.5.5 Evaluation of Design Options 92
4.5.6 Comparison with State-of-the-Art 94
4.5.7 Extensibility as A Saliency Optimization Algorithm 108
4.5.8 Efficiency 111
4.5.9 Limitation 111
4.6 Summary 111
5 DNN Based Saliency Detection 113
5.1 Problem Formulation 114
5.2 Contributions 115
5.3 Related Works 117
5.3.1 Auto-Encoder 117
5.4 Saliency Detection with Adaptive Background Search and Foreground Estimation (BSFE) Using Comprehensive Auto-Encoder 118
5.4.1 Adaptive Background Search 119

5.4.2 Foreground Estimation 120
5.5 Experimental Results of BSFE 124
5.5.1 Datasets 124
5.5.2 Evaluation Metrics 124
5.5.3 Parameters 124
5.5.4 Implementation 125
5.5.5 Evaluation Against State-of-the-Art 125
5.6 Saliency Detection with Multi-Dimensional Features Using DNN Based Dense and Sparse Labeling (DSL) 134
5.6.1 Dense Labeling for Initial Saliency Estimation 136
5.6.2 Sparse Labeling for Initial Saliency Estimation 139
5.6.3 Sparse Labeling for Final Saliency Map 142
5.7 Experimental Results of DSL 146
5.7.1 Datasets 147
5.7.2 Evaluation Metrics 148
5.7.3 Implementation 149
5.7.4 Parameter Analysis of the DL Step 149
5.7.5 Parameter Analysis of the SL Step 150
5.7.6 Parameter Analysis of the DC Step 152
5.7.7 Contribution Comparison 152
5.7.8 Evaluation Against Conventional Methods 154
5.7.9 Evaluation Against Learning Based Methods 159
5.7.10 Efficiency 165
5.7.11 Limitation 165
5.8 Summary 166

6 Conclusions and Future Works 167
6.1 Conclusions 167
6.2 Future Works 169
References 171
Appendix A 194

List of Figures

FIGURE 1.1 EXAMPLES OF SALIENT OBJECTS IN NATURAL IMAGES.. 23

FIGURE 1.2 EXAMPLES SHOWING THE PROBLEMATIC PRE-ASSUMPTIONS IN CONVENTIONAL LOW-LEVEL FEATURE BASED SALIENCY DETECTION METHODS. 26

FIGURE 1.3 THE CHALLENGES REGARDING LOW CONTRAST IMAGES AND COMPLEX IMAGES 28

FIGURE 2.1 EXAMPLE SALIENCY MAPS OF PREVALENT BOTTOM-UP SALIENCY DETECTION METHODS 37

FIGURE 2.2 EXAMPLE SALIENCY MAPS OF PREVALENT TOP-DOWN SALIENCY DETECTION METHODS 40

FIGURE 2.3 ILLUSTRATION OF THE DIFFERENCE BETWEEN SALIENCY DETECTION AND GENERAL IMAGE SEGMENTATION. 42

FIGURE 2.4 AN ILLUSTRATIVE DIAGRAM OF A "NEURON" IN DNN. 47

FIGURE 2.5 A SMALL ILLUSTRATIVE NEURAL NETWORK WITH 3 LAYERS. 49

FIGURE 2.6 A TYPICAL CNN ARCHITECTURE. 50

FIGURE 3.1 THE EFFECT OF ERRONEOUS BOUNDARY REMOVAL IN SECTION 3.4.1. 58

FIGURE 3.2 EXAMPLES THAT (3.16) LEADS TO MORE PRECISE SALIENCY OUTPUTS. 62

FIGURE 3.3 PRECISION-RECALL CURVES ON THE MSRA10K DATASET WITH DIFFERENT DESIGN OPTIONS OF THE PROPOSED APPROACH. 66

FIGURE 3.4 PRECISION-RECALL CURVES (PART 1) OF DIFFERENT METHODS ON THE MSRA10K DATASET. 67

FIGURE 3.5 PRECISION-RECALL CURVES (PART 2) OF DIFFERENT METHODS ON THE MSRA10K DATASET. 68
FIGURE 3.6 AVERAGE F-MEASURES OF DIFFERENT METHODS ON THE MSRA10K DATASET. 68
FIGURE 3.7 SALIENCY MAP EXAMPLES OF DIFFERENT METHODS ON THE MSRA10K DATASET 69
FIGURE 3.8 PRECISION-RECALL CURVES (PART 1) OF DIFFERENT METHODS ON THE DUT-OMRON DATASET. 71
FIGURE 3.9 PRECISION-RECALL CURVES (PART 2) OF DIFFERENT METHODS ON THE DUT-OMRON DATASET. 71
FIGURE 3.10 AVERAGE F-MEASURES OF DIFFERENT METHODS ON THE DUT-OMRON DATASET. 72
FIGURE 3.11 SALIENCY MAP EXAMPLES OF DIFFERENT METHODS ON THE DUT-OMRON DATASET 73
FIGURE 4.1 EXAMPLES SHOWING THE PROBLEM OF USING BOUNDARIES AS BACKGROUND QUERIES WHEN THE SALIENT OBJECTS ARE BOUNDARY-ADJACENT 78
FIGURE 4.2 EXAMPLES OF RC 84
FIGURE 4.3 EXAMPLES OF RRWR 87
FIGURE 4.4 AVERAGE F-MEASURES WITH DIFFERENT $t_{reverse}$ USED IN RC ON THE MSRA10K DATASET 91
FIGURE 4.5 AVERAGE F-MEASURES WITH DIFFERENT η USED IN RRWR ON THE MSRA10K DATASET 92

FIGURE 4.6 THE PRECISION-RECALL CURVES OF OUR METHOD, OUR METHOD WITHOUT USING RC, AND OUR METHOD WITHOUT USING RRWR.....93
FIGURE 4.7 THE AVERAGE F-MEASURES OF OUR METHOD, OUR METHOD WITHOUT USING RC, AND OUR METHOD WITHOUT USING RRWR. 94
FIGURE 4.8 PRECISION-RECALL CURVES ON THE MSRA10K DATASET. 96
FIGURE 4.9 F-MEASURES ON THE MSRA10K DATASET. 96
FIGURE 4.10 MAE SCORES ON THE MSRA10K DATASET. 97
FIGURE 4.11 PRECISION-RECALL CURVES ON THE ECSSD DATASET.....98
FIGURE 4.12 F-MEASURES ON THE ECSSD DATASET. 98
FIGURE 4.13 MAE SCORES ON THE ECSSD DATASET. 99
FIGURE 4.14 PRECISION-RECALL CURVES ON THE SED DATASET.....100
FIGURE 4.15 F-MEASURES ON THE SED DATASET. 100
FIGURE 4.16 MAE SCORES ON THE SED DATASET. 101
FIGURE 4.17 PRECISION-RECALL CURVES ON THE PASCAL-S DATASET. 101
FIGURE 4.18 F-MEASURES ON THE PASCAL-S DATASET.....102
FIGURE 4.19 MAE SCORES ON THE PASCAL-S DATASET. 102
FIGURE 4.20 PRECISION-RECALL CURVES ON THE BAOS DATASET. 103
FIGURE 4.21 F-MEASURES ON THE BAOS DATASET. 104
FIGURE 4.22 MAE SCORES ON THE BAOS DATASET.....104
FIGURE 4.23 SALIENCY MAP EXAMPLES OF STATE-OF-THE-ART METHODS AGAINST OUR RCRR METHOD 107
FIGURE 4.24 OPTIMIZATION EVALUATION RESULTS OF F-MEASURE ON THE MR METHOD. 109
FIGURE 4.25 OPTIMIZATION EVALUATION RESULTS OF MAE ON THE MR METHOD. 109

FIGURE 4.26 OPTIMIZATION EVALUATION RESULTS OF F-MEASURE ON THE MC METHOD. 110
FIGURE 4.27 OPTIMIZATION EVALUATION RESULTS OF MAE ON THE MC METHOD. 110
FIGURE 4.28 EXAMPLE CASE SHOWING THE LIMITATION OF OUR PROPOSED RCRR METHOD 111
FIGURE 5.1 ILLUSTRATION OF CHALLENGES ENCOUNTERED BY CONVENTIONAL LOW-LEVEL FEATURE BASED SALIENCY DETECTION METHODS 115
FIGURE 5.2 FLOWCHART OF OUR PROPOSED BSFE METHOD. 119
FIGURE 5.3 EXAMPLES OF THE BACKGROUND MASK BY THE BS SAE MODEL. 120
FIGURE 5.4 FLOWCHART OF OUR PROPOSED BSFE METHOD. 123
FIGURE 5.5 PRECISION-RECALL CURVES ON THE ECSSD DATASET. 126
FIGURE 5.6 F-MEASURES ON THE ECSSD DATASET. 126
FIGURE 5.7 MAE SCORES ON THE ECSSD DATASET. 127
FIGURE 5.8 PRECISION-RECALL CURVES ON THE PASCAL-S DATASET. 127
FIGURE 5.9 F-MEASURES ON THE PASCAL-S DATASET. 128
FIGURE 5.10 MAE SCORES ON THE PASCAL-S DATASET. 128
FIGURE 5.11 PRECISION-RECALL CURVES ON THE SED1 DATASET. 129
FIGURE 5.12 F-MEASURES ON THE SED1 DATASET. 129
FIGURE 5.13 MAE SCORES ON THE SED1 DATASET. 130
FIGURE 5.14 PRECISION-RECALL CURVES ON THE SED2 DATASET. 130
FIGURE 5.15 F-MEASURES ON THE SED2 DATASET. 131
FIGURE 5.16 MAE SCORES ON THE SED2 DATASET. 131
FIGURE 5.17 SALIENCY MAP EXAMPLES OF STATE-OF-THE-ART METHODS AGAINST OUR BSFE METHOD 133

FIGURE 5.18 FLOWCHART OF OUR DSL METHOD 135
FIGURE 5.19 FLOWCHART OF THE DL STEP. 136
FIGURE 5.20 BILINEAR INTERPOLATION FROM THE CONV8 LAYER TO THE DECONV32 LAYER. 138
FIGURE 5.21 EXAMPLE OUTPUTS OF THE DL STEP 139
FIGURE 5.22 FLOWCHART OF THE SL STEP 140
FIGURE 5.23 EXAMPLE OUTPUTS OF THE DL, SL, AND DC STEPS 146
FIGURE 5.24 PRECISION-RECALL CURVES AGAINST CONVENTIONAL METHODS ON THE ECSSD DATASET. 155
FIGURE 5.25 PRECISION-RECALL CURVES AGAINST CONVENTIONAL METHODS ON THE PASCAL-S DATASET. 156
FIGURE 5.26 PRECISION-RECALL CURVES AGAINST CONVENTIONAL METHODS ON THE SED1 DATASET 156
FIGURE 5.27 PRECISION-RECALL CURVES AGAINST CONVENTIONAL METHODS ON THE SED2 DATASET 157
FIGURE 5.28 PRECISION-RECALL CURVES AGAINST CONVENTIONAL METHODS ON THE THUR15K DATASET. 157
FIGURE 5.29 PRECISION-RECALL CURVES AGAINST CONVENTIONAL METHODS ON THE HKU-IS DATASET 158
FIGURE 5.30 PRECISION-RECALL CURVES AGAINST LEARNING BASED METHODS ON THE ECSSD DATASET. 160
FIGURE 5.31 PRECISION-RECALL CURVES AGAINST LEARNING BASED METHODS ON THE PASCAL-S DATASET. 161

FIGURE 5.32 PRECISION-RECALL CURVES AGAINST LEARNING BASED METHODS ON THE SED1 DATASET.....161
FIGURE 5.33 PRECISION-RECALL CURVES AGAINST LEARNING BASED METHODS ON THE SED2 DATASET.....162
FIGURE 5.34 PRECISION-RECALL CURVES AGAINST LEARNING BASED METHODS ON THE THUR15K DATASET. 162
FIGURE 5.35 PRECISION-RECALL CURVES AGAINST LEARNING BASED METHODS ON THE HKU-IS DATASET.....163
FIGURE 5.36 SALIENCY MAP EXAMPLES OF STATE-OF-THE-ART METHODS AGAINST OUR DSL METHOD 164

List of Tables

TABLE 2.1 INFORMATION STATISTICS OF BOTTOM-UP SALIENCY DETECTION METHODS 36

TABLE 2.2 INFORMATION STATISTICS OF TOP-DOWN SALIENCY DETECTION METHODS.....39

TABLE 3.1 ALGORITHM DESCRIPTION OF OUR PROPOSED RR METHOD.....63

TABLE 3.2 RUNNING TIME TEST RESULTS OF SELECTED METHODS (SECONDS PER IMAGE).74

TABLE 4.1 ALGORITHM DESCRIPTION OF THE RC PROCESS 82

TABLE 4.2 ALGORITHM DESCRIPTION OF OUR PROPOSED RCRR METHOD 87

TABLE 4.3 F-MEASURE AND MAE EVALUATION RESULTS.....95

TABLE 5.1 ALGORITHM DESCRIPTION OF OUR PROPOSED FOREGROUND ESTIMATION.....123

TABLE 5.2 HYPER-PARAMETERS FOR THE TRAINING OF BS SAE AND FE SAE.....125

TABLE 5.3 ARCHITECTURE OF OUR DL NETWORK.....137

TABLE 5.4 ARCHITECTURE OF OUR LOCAL CNN.....141

TABLE 5.5 ARCHITECTURE OF OUR DC NETWORK 143

TABLE 5.6 PERFORMANCES OF THE DL NETWORK AGAINST TWO STATE-OF-THE-ART DENSE LABELING MODELS.....150

TABLE 5.7 PERFORMANCES OF THE SL NETWORK WITH DIFFERENT LAYER NUMBER (#LAYER) AND PARAMETERS PER LAYER (#PARAM) 151

TABLE 5.8 PERFORMANCES OF DSL WITH DIFFERENT SL FEATURE COMBINATIONS 151

TABLE 5.9 PERFORMANCES OF THE DC STEP WITH DIFFERENT BASELINE MODELS ON THE TWO CHALLENGING DATASETS ECSSD AND PASCAL-S 152

TABLE 5.10 PERFORMANCES OF DIFFERENT DESIGN OPTION CONFIGURATIONS ON THE TWO CHALLENGING DATASETS ECSSD AND PASCAL-S 153

TABLE 5.11 QUANTITATIVE EVALUATION RESULTS OF DSL AGAINST CONVENTIONAL SALIENCY DETECTION METHODS 158

TABLE 5.12 QUANTITATIVE EVALUATION RESULTS OF DSL AGAINST LEARNING BASED SALIENCY DETECTION METHODS 163

TABLE 5.13 EFFICIENCY COMPARISON (SECONDS PER IMAGE) 165

Chapter 1 Introduction

1.1 Background of Saliency Detection

With the rapid uptake of smart devices and social networks, we are now immersed in massive amounts of digital media data every day. Considering the scarcity of our attention and time, it is urgent and advantageous to filter out only the most useful message for further processing among all of the available data to us. This concept equates to the saliency detection process when applied to images.

Saliency is usually referred to as local contrast [5-7], which typically originates from contrasts between objects and their surroundings, such as differences in color, texture, shape, etc. This mechanism measures intrinsically salient stimuli to the vision system that primarily attracts human attention in the early stage of visual exposure to an input image [6]. Intermediate and higher visual processes may automatically judge the importance of different regions of the image, and conduct detailed processes only on the "salient object" that mostly related to the current task, while neglecting the remaining "background" regions [8]. Figure 1.1 shows a few examples of natural images. As seen in Figure 1.1(c), the flower, the cookies, the girl, the cat and the toy car usually attract the most visual attention in their corresponding images, and thus are regarded as salient objects. On the other hand, Figure 1.1(b) shows illustrative results of saliency detection, or the "saliency maps" in formal terms. The general objective of saliency detection is to provide saliency maps of the input images as close to the ground truth as possible.

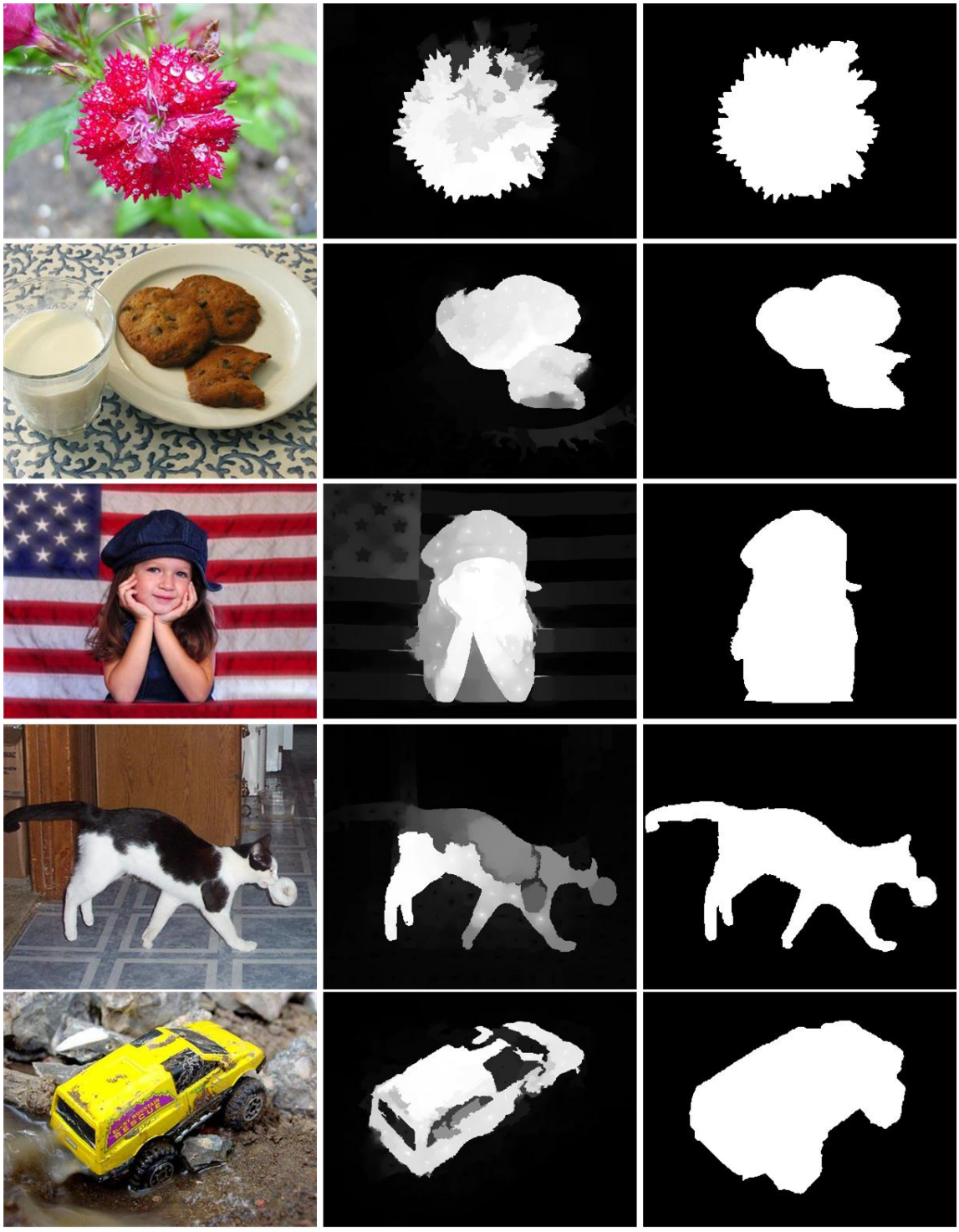

Figure 1.1 Examples of salient objects in natural images. (a) original images; (b) example saliency detection results [2]; (c) ground truth.

1.2 Significance of Research

Human visual saliency detection has long been studied by cognition scientists and has recently draw much of interest in the computer vision community mainly because of its assistance in finding the objects or regions that efficiently represent a scene, and thus harness complex vision problems such as scene understanding. Early researches of saliency detection mostly focus on human eye fixation [5], [9], [10], which approximates the visual attention of semantic objects in a given image, such as human faces, texts, or daily objects [9], [11]. The detection results of eye fixations, however, are often presented as sparse dots without details about the objects. On the other hand, the recent researches of saliency detection are capable of locating and segmenting the whole salient object with complete boundary details [12], and thus has received broad research interests. The detection of the salient objects in images is of significant importance, as it not only improves the subsequent image processing and analyses, but also directs the limited computational resources to more efficient solutions. Saliency detection has received recognized success in various areas, such as computer vision, graphics, and robotics. More specifically, the proposed models have been broadly applied in object detection and recognition [13-20], object discovery [21], [22], photo collage and thumbnailing [23-25], image quality assessment [26-28], image segmentation [29-32], content based image retrieval [21], [33-35], image editing and manipulating [30], [36-38], image and video compression [39], [40], video summarization [41-43], visual tracking [28], [44-49], and human-robot interaction [50], [51].

1.3 Existing Challenges

Since emergence, intensive researches have been conducted on saliency detection. The majority of existing saliency detection methods is based on hand-crafted low-level features. However, there are multiple critical issues on the existing methods that prevent them from perfection.

1.3.1 Problematic Pre-assumptions

Among many conventional low-level feature based saliency detection methods, specific pre-assumptions or prior knowledge are required in order to make them properly functioning. Most of the pre-assumptions are largely empirical, e.g. image boundary regions are assumed as background [52], [53], or image central [54], [55] regions are assumed as foreground. These pre-assumptions are easily violated on broader datasets with more unusual-patterned images, such as the example in Figure 1.2, where the upper two images have salient objects on the boundary, while the lower two images have background regions in the center. The atypical patterns of these images lead to the failure of conventional low-level feature based methods, as seen in Figure 1.2(b). To overcome the limitations above, multiple more robust improvements of the pre-assumptions have been proposed, which will be discussed in Chapters 3 and 4.

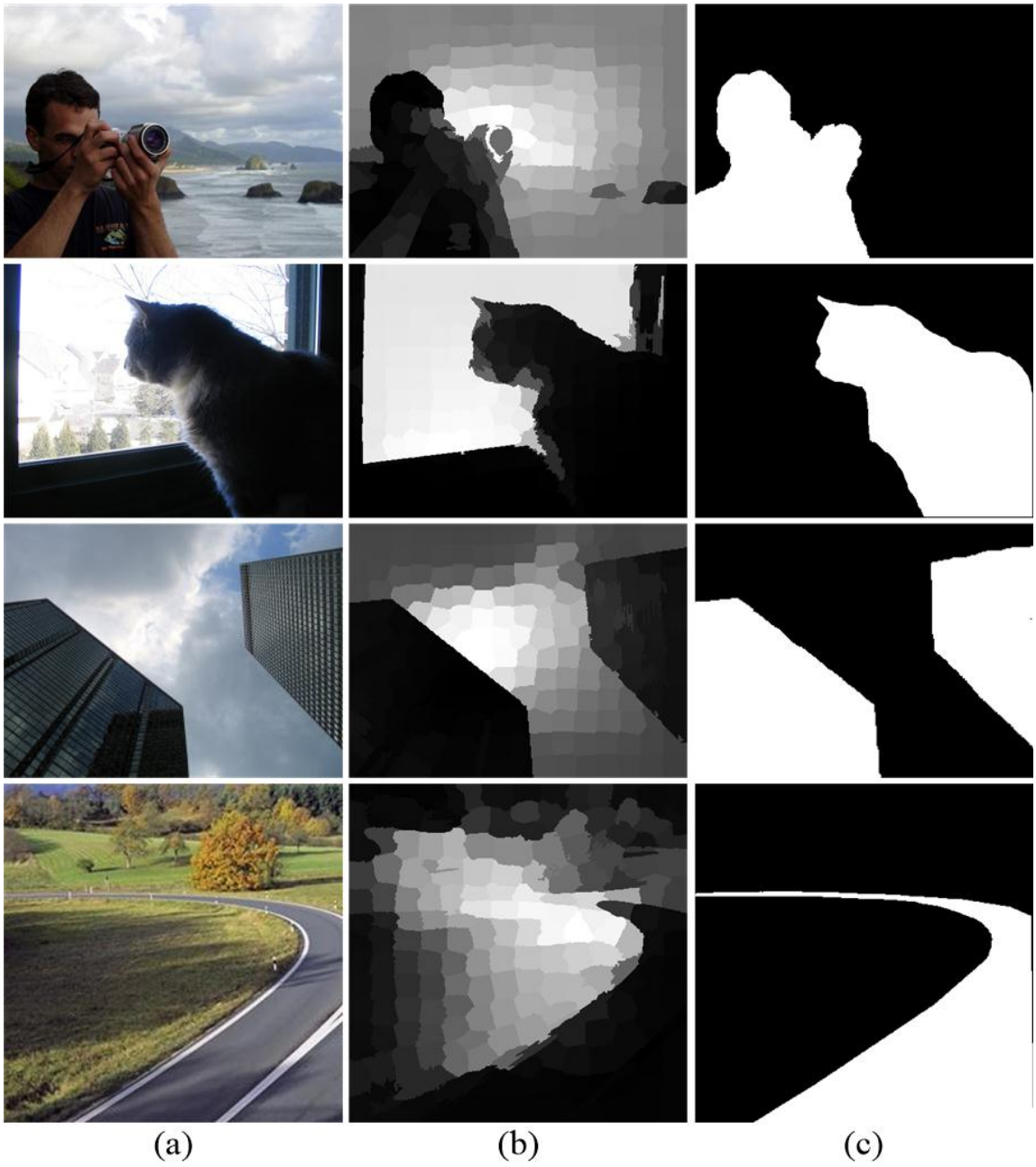

Figure 1.2 Examples showing the problematic pre-assumptions in conventional low-level feature based saliency detection methods. (a) original images; (b) failed detection results by a conventional low-level feature based method [52]; (c) ground truth.

1.3.2 Ineffective Feature Integration

Among the hand-crafted low-level features in conventional methods, each one is usually advantageous only on a specific aspect, e.g. color histogram is good at differentiating texture patterns, frequency spectrum is good at differentiating energy patterns, and SIFT

[56] is good at object recognition with varied environments, etc. It is generally difficult to combine different low-level features into a single algorithm to benefit from them all. Although some integration trials have been made [57], [58], these specially designed algorithms are nevertheless bulky and inefficient due to the large number of features involved. On the other hand, a more effective means of feature integration has been proposed, which will be discussed in Chapter 5.

1.3.3 Absence of High-Level Feature Abstraction and Learning

Without feature abstraction and learning, the conventional low-level feature based methods are likely to encounter difficulty regarding low contrast images and complex patterned images. Some typical examples of this issue are exhibited in Figure 1.3, where the upper two images lead to the failed results on low contrast images, while the lower two images lead to the failed results on complex patterned images. On the other hand, however, this drawback can be readily solved via high-level feature extraction and learning. The recently prevalent deep neural networks (DNNs), especially the convolutional neural networks (CNNs), are proved to be of great assistance in high-level feature extraction of saliency detection. This will be discussed in Chapter 5.

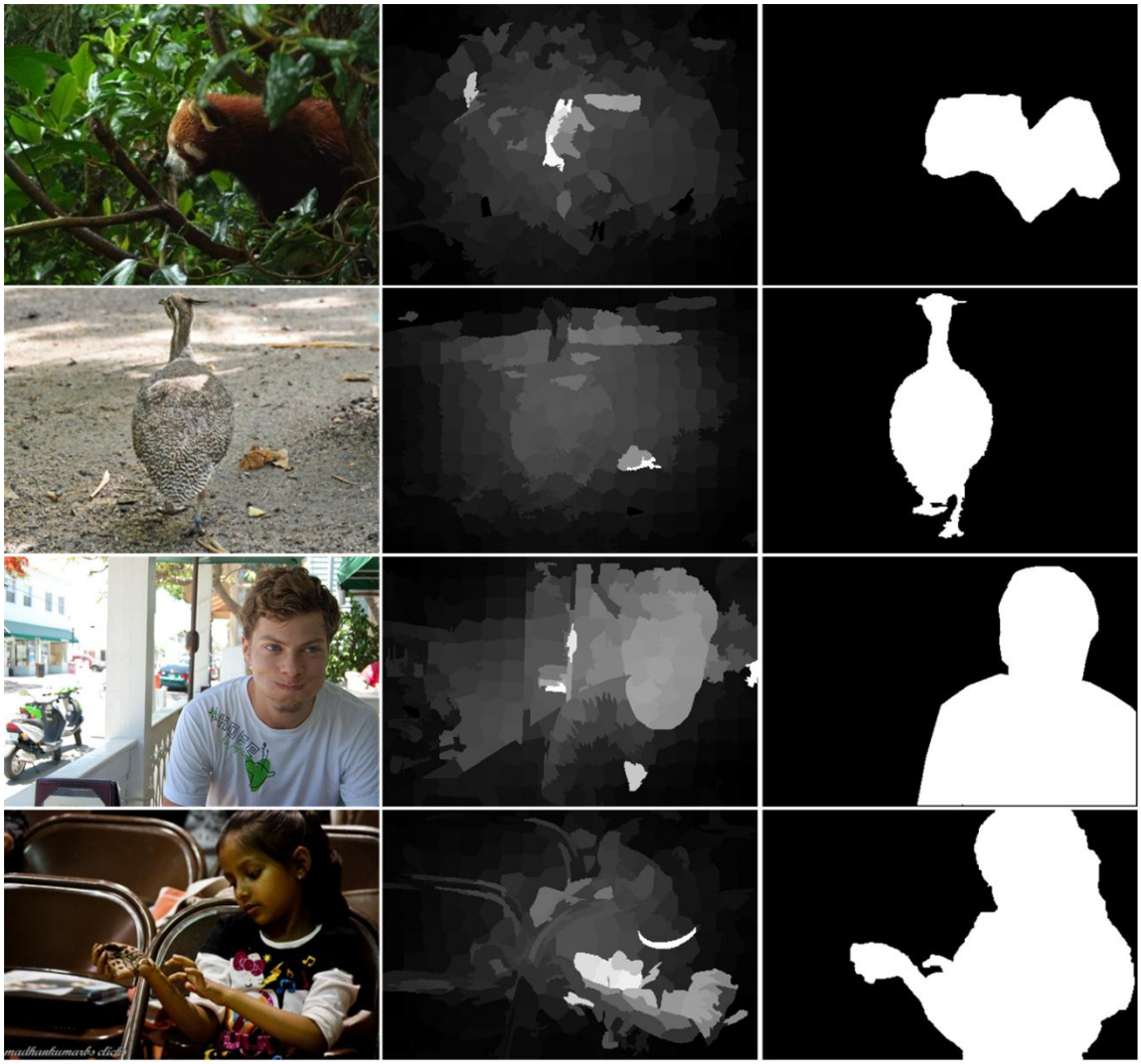

Figure 1.3 The challenges regarding low contrast images and complex images. (a) original images; (b) failed detection results by a conventional low-level feature based method [54]; (c) ground truth.

1.4 Contributions

To address the issues above in existing saliency detection methods, we have conducted extensive research on three major aspects, and have proposed four novel saliency detection methods to provide improved saliency detection performances. The major contributions are summarized below.

1.4.1 Conventional Low-Level Feature Based Saliency Detection

We first explore better exploitations of the hand crafted features of conventional low-level feature based saliency detection methods, and propose the regularized random walks ranking (RR) method, which has the following contributions:

(1) To improve the background saliency estimation, we first filter out one of the four boundaries of the input image that most unlikely belong to the background, unlike conventional methods that use all four boundaries as background reference [52], [53]. This erroneous boundary removal process effectively eliminates the image boundary with boundary-adjacent foreground superpixels, and thus neutralizes their negative influences in the saliency estimations.

(2) To improve the foreground saliency estimation, we propose the regularized random walks ranking algorithm, which consists of a pixel-wise graph term and a newly formulated fitting constraint to take local image data and prior estimation into account. This fitting constraint is able to utilize the entire saliency estimation results from the former steps instead of the selected seed points alone. Besides, regularized random walks ranking is independent of superpixel segmentation, and can generate pixel-wised saliency maps that reflect full-details of the input image.

The RR method has been published on CVPR 2015 [1], and will be fully described in Chapter 3.

1.4.2 Improved Low-Level Feature Based Saliency Detection

To improve the performance of the RR method in Section 1.4.1, we have conducted further research about the boundary regions in an image, and propose the reversion

correction and regularized random walks ranking (RCRR) method, which is a direct upgrade of the RR method. RCRR has the following contributions:

(1) We propose the reversion correction (RC) process, which, unlike the RR method that completely removes one of the problematic boundaries, locates and eliminates the boundary-adjacent foreground superpixels, which is more accurate and can maximally preventing the saliency reversions (will be discussed later) from emerging. This mechanism also leads to increased robustness of the algorithm.

(2) We present the extensibility of our method as a saliency optimization algorithm, which can be directly applied on existing saliency detection methods for performance improvement purposes.

(3) We also propose the boundary-adjacent object saliency (BAOS) dataset, which is comprised of 200 images that have large proportions of the salient objects on the image boundaries. This dataset provides an objective evaluation for saliency detection methods' performance on boundary-adjacent salient objects.

The RCRR method has been publish on IEEE TIP [2], and will be fully described in Chapter 4.

1.4.3 Deep Neural Network Based Saliency Detection

Among various recent research works in computer vision, the deep neural network (DNN) [59] has shown particular success in high-level feature extraction, which grants us an excellent machine learning tool to overcome the difficulty of conventional low-level feature based saliency detection methods when facing low contrast images and complex patterned images. We propose two independent DNN based methods, the

adaptive background search and foreground estimation (BSFE) and the dense and sparse labeling (DSL). The contributions of the two methods are listed below.

For BSFE:

(1) We propose an adaptive background extractor, which approximates background regions semantically and cognitively, contributing to higher detection accuracy;

(2) We apply the auto-encoder (AE) hierarchically for foreground estimation, which is guided by the background mask, to reconstruct the final saliency map with higher performance.

And for DSL:

(1) We combine the DNN-based dense labeling (DL) and sparse labeling (SL) together for initial saliency estimation, in which DL conducts dense labeling that maximally preserves the global image information and provides accurate location estimation of the salient object, while SL conducts sparse labeling that focuses more on local features of the salient object;

(2) For the SL step, both low-level features and RGB features of the image are applied as the network inputs. Such multi-dimensional input features enable the complementary advantage of low-level features and RGB features, by which the image is more accurately abstracted and represented;

(3) In the last deep convolution (DC) step, a 6-channeled input structure is proposed, which provides significantly better guidance in generating the final saliency map. On the one hand, the combined initial saliency estimations from the DL and SL steps provide accurate location guidance of the salient object, effectively excluding any false salient region; on the other hand, the superpixel indication channel precisely represents the

current to-be-classified superpixel, which leads to more consistent and accurate saliency labeling.

The BSFE method has been published on ICIP 2016 [3], while the DSL method has been published on IEEE TCSVT [4]. They will be fully described in Chapter 5.

1.5 Thesis Organization

The remainder of this thesis is organized as follows.

Chapter 2 gives a systematic review of the related works, including the categorization of saliency detection methods, and the different prevalent applications of DNN.

Chapter 3 introduces the RR method in detail, which includes research objective, necessary prior knowledge (manifold ranking and random walks), step-by-step methodology (background/foreground saliency estimation, and final saliency formulation), and experimental results.

Chapter 4 introduces the RCRR method in detail, which includes research objective, necessary prior knowledge (*k*-means clustering), step-by-step methodology (reversion correction and regularized random walks ranking), and experimental results.

Chapter 5 introduces the two DNN based methods, i.e. BSFE and DSL. The research objectives and related works (auto-encoder, sparse labeling and dense labeling) of the two methods are first presented, followed by the methodology (adaptive background search and foreground estimation) and experimental results of BSFE, and then the detailed description of DSL (dense labeling, sparse labeling and deep convolution) and its experimental evaluations.

Chapter 6 summarizes the whole thesis, gives conclusions, and explores for potential future works.

Chapter 2 Related Works

In this chapter, we will systematically review the related works about this thesis, namely the categorizations of saliency detection methods, developments of object detection, and the most prevalent applications of deep neural networks.

2.1 Saliency Detection

From the perspective of computer vision, the methods of saliency detection are broadly categorized into two major groups, namely the bottom-up methods and the top-down methods. Besides that, more methods using unconventional models and features have also been proposed in recent years.

2.1.1 Bottom-Up Methods

The bottom-up methods are largely designed for non-task-specific saliency detections [60], in which low-level features are mainly involved as fundamentals for the detections. These features are usually data-driven and hand-crafted.

Before the 2010s, the researches of saliency detection are in the stage of fundamental developments, which draws interest across multiple disciplines including cognitive psychology, neuroscience, and computer vision. At this time, usually only the most basic features in conventional image processing, such as pixel color value, histogram, frequency spectrum, etc., are exploited in the methods. As a pioneer, Itti *et al.* [5] present a center-surround model that integrates color, intensity and orientation at different scales for saliency detection. Rahtu *et al.* [61] detect saliency by measuring the

center-surround contrast of a sliding window over the input image. Bruce *et al.* [62] exploit Shannon's self-information measurement on local context to compute saliency. In the work of Cheng *et al.* [63], [64], pixel-wise color histogram and region-based contrast are utilized in establishing the histogram-based and region-based saliency maps. Duan *et al.* [65] measure global contrast based saliency with spatially weighted feature dissimilarities. Achanta *et al.* [66] propose a frequency-tuned method based on color and luminance, in which the saliency value is computed by the color difference with respect to the mean pixel value. Fourier spectrum analysis has also been utilized in visual saliency detection, such as in the works of Hou *et al.* [67] and Guo *et al.* [68].

Since the 2010s, more advanced models, and especially the graph based models, have been introduced to saliency detection, which have greatly improved the overall detection accuracy. It is also notable that the majority of conventional low-level feature based saliency detection methods were proposed during this period. Jiang *et al.* [54] establish a 2-ring graph model that calculates saliency values of different image regions by their Markov absorption probabilities. To overcome the negative influence of small-scale high-contrast image patterns, Yan *et al.* [69] propose a multi-layer approach that optimizes saliency detection by a hierarchical tree model. Perazzi *et al.* [70] unify the contrast and saliency computation into a single high dimensional Gaussian filtering framework. Wei *et al.* [71] apply background priors and geodesic distance to compute visual saliency. Yang *et al.* [52] exploit the graph-based manifold ranking in extracting foreground queries for the final saliency map, in which the four image boundaries are used as background prior knowledge. In the work of Li *et al.* [1], the image boundaries are refined before being used as background prior knowledge, and a random-walk based ranking model is applied for saliency optimization. And in the work of Qin *et al.* [72],

the saliency of different image cells is computed by synchronous update of their dynamic states via the cellular automata model.

The statistics of prevalent bottom-up methods are listed in Table 2.1, and some example saliency maps are shown in Figure 2.1. It is observed that the bottom-up methods generally behave poorly on low contrast or complex patterned images.

Table 2.1 Information statistics of bottom-up saliency detection methods

#	Method	Published on	Year	Code
1	IT [5]	TPAMI	1998	M
2	SR [67]	CVPR	2007	M
3	SUN [73]	JOV	2008	M
4	FT [66]	CVPR	2009	C
5	SEG [61]	ECCV	2010	M+C
6	RC/HC [63]	CVPR	2011	C
7	SVO [74]	ICCV	2011	M+C
8	CB [75]	BMVC	2011	M+C
9	FES [76]	IA	2011	M+C
10	SF [70]	CVPR	2012	C
11	LR [77]	CVPR	2012	M
12	CA [78]	CVPR	2012	M+C
12	PCA [79]	CVPR	2013	M+C
13	HS [69]	CVPR	2013	EXE
14	MR [52]	CVPR	2013	M
15	MC [54]	ICCV	2013	M+C
16	DSR [80]	ICCV	2013	M+C
17	GC [81]	ICCV	2013	C
18	UFO [82]	ICCV	2013	M+C
19	GR [83]	SPL	2013	M+C
20	RBD [53]	CVPR	2014	M
21	RR [1]	CVPR	2015	M
22	BSCA [72]	CVPR	2015	M
23	RCRR [2]	TIP	2016	M

Abbreviation of journals and conferences: please refer to Appendix A; Abbreviation of code type: M - Matlab; C - C/C++; EXE - executable.

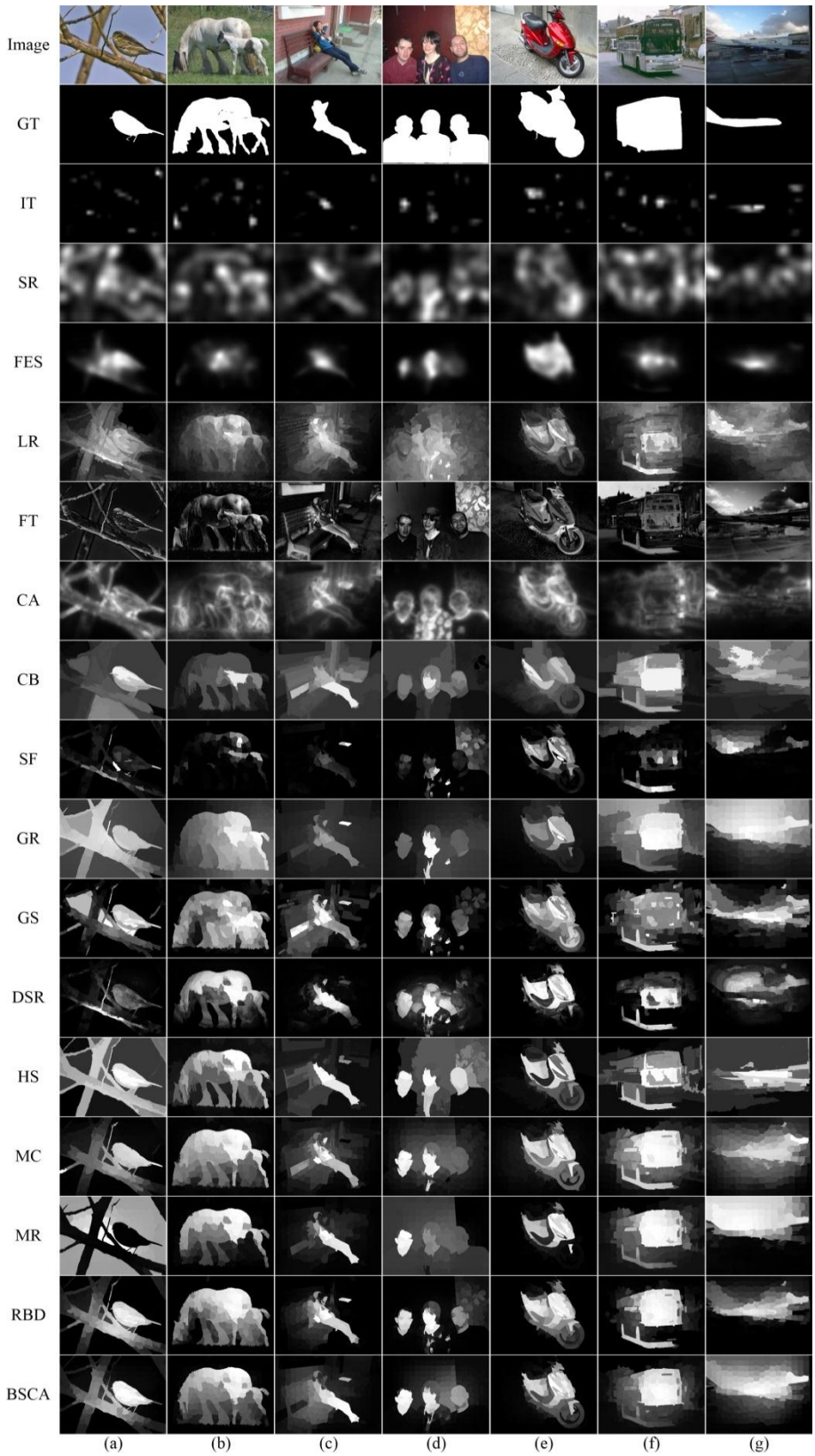

Figure 2.1 Example saliency maps of prevalent bottom-up saliency detection methods. (a) – (g): image case IDs.

2.1.2 Top-Down Methods

On the other hand, the top-down saliency detection methods are usually task-driven. These methods break down the saliency detection task into more fundamental components, and task-specific high-level features are frequently involved as prior knowledge. Supervised learning approaches are commonly used in detecting image saliency. In the work of Yang *et al.* [84], joint learning of conditional random field (CRF) is conducted in discriminating visual saliency. Lu *et al.* [85] apply a graph-based diffusion process to learn the optimal seeds of an image to discriminate object and background. Mai *et al.* [86] train a CRF model to aggregate saliency maps from various models, which benefits not only from the individual saliency maps, but also from the interactions among different pixels. And in the work of Tong *et al.* [87], samples from a weak saliency map are exploited as the training set for a series of supply vector machines (SVMs) [88], which are subsequently applied to generate a strong saliency map.

Since 2013, benefitted from the tremendous success of deep learning and other high-level feature extraction techniques, more learning based methods arise with significantly improved performances. Jiang *et al.* [57] regard saliency detection as a regression problem, which fuses regional contrast, property and backgroundness into a random forest classifier for multi-level image saliency segmentation. Kim *et al.* [89] represent the saliency map as a linear combination of different high-dimensional color space, where the salient regions and the background distinctively separated. Wang *et al.* [90] train two separate DNNs with image patches (DNN-L) and object proposals (DNN-G) for local and global saliency, the two results are then integrated by a weighted

summation to create the final saliency map. Zhao *et al.* [91] establish a multi-context DNN model for superpixel-wise saliency classification, which exploits DNN for per-superpixel saliency value classification. Li *et al.* [92] propose a similar multi-scale DNN model for feature extraction, the outputs of which are then aggregated for the final saliency map. And in the work of Chen *et al.* [93], two stacked DNNs are utilized to build the saliency detection model, among which the first one provides a coarse saliency estimation with the whole image as input, while the second one focuses on the local context to produce fine-grained saliency map.

The statistics of popular top-down saliency detection methods are listed in Table 2.2, and some example saliency maps are shown in Figure 2.2. We notice that compared with the bottom-up methods in Figure 2.1, the top-down methods generally perform much better on low contrast and complex patterned images, which is attributed to the high-level feature extraction involved in their learning processes.

Table 2.2 Information statistics of top-down saliency detection methods.

#	Method	Published on	Year	Code
1	SA [86]	CVPR	2013	M+C
2	DRFI [57]	CVPR	2013	M+C
3	HDCT [89]	CVPR	2014	M
4	BL [87]	CVPR	2015	M
5	MCDL [91]	CVPR	2015	Py+C
6	LEGS [90]	CVPR	2015	M+C
7	MDF [92]	CVPR	2015	M+C
8	DISC [93]	TNNLS	2015	M+C
9	DSL [4]	TCSVT	2016	M

Abbreviation of journals and conferences: please refer to Appendix A; Abbreviation of code type: M - Matlab; C - C/C++; Py - Python.

Figure 2.2 Example saliency maps of prevalent top-down saliency detection methods. (a) – (g): image case IDs.

2.1.3 Other Methods

In recent years, more applications of other innovative models and features have been proposed in saliency detection. For instance, with the application of commercial plenoptic cameras, Li et al. [94] propose a saliency detection method which exploits the unique refocusing capability of light fields. Liu et al. [95] design an adaptive partial differential equation (PDE) system learning from images, which is used to model the evolution of visual saliency. Yang *et al.* [96] establish a visual tracking model of the salient object based on midlevel structural information captured in superpixels. The work of Zhou et al. [97] develops the time-mapping model, which is a time-based spatial

tone-mapping that is used to convert high-frame-rate video into low-frame-rate video while maximally preserving the saliency information contained. In the work of Vig et al. [98], they propose the hierarchical feature learning method using a data-driven approach to perform large-scale searching of optimal features; this method provide integrated and biologically-plausible saliency detection outputs. The accuracy and robustness of the methods above, however, are still under further validation.

2.2 Image Segmentation

Image segmentation, which includes semantic scene labeling and semantic segmentation, is one of the well-developed research areas in computer vision [99]. While saliency detection aims to locate the most salient object in an image, and treat the segmentation task as a binary labeling problem, the objective of general image segmentation is to mark each pixel in the image a label indicating the type of object class it belongs to (background is treated as a separate class in this case). In other words, image segmentation is a multi-class labeling problem. Figure 2.3 shows typical examples to illustrate the difference between saliency detection and general image segmentation.

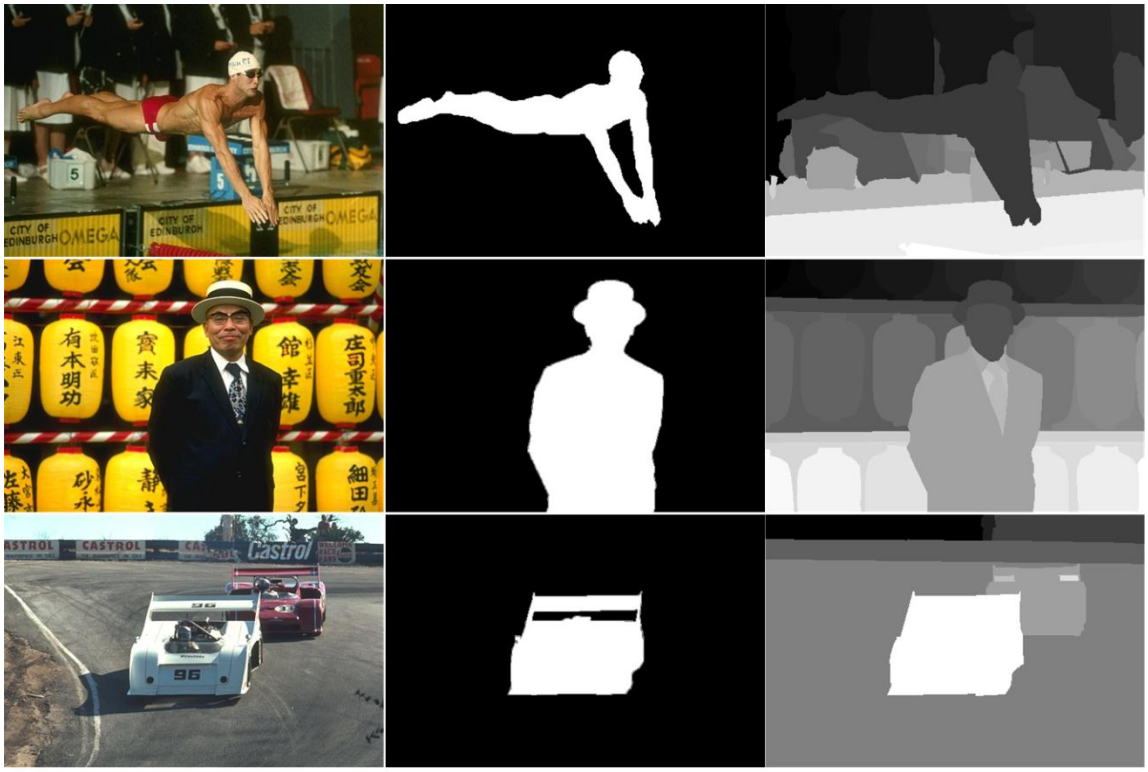

Figure 2.3 Illustration of the difference between saliency detection and general image segmentation. (a) original images; (b) ground truth for saliency detection; (c) ground truth for multi-class image segmentation.

The segmentation methods are usually categorized as unsupervised methods and supervised methods. Reviews of both types can be found in [100-102]. Unsupervised methods are conducted without any prior knowledge or user input, which encourages their significantly high efficiency. These methods include thresholding [103], [104], relaxation [105], edge detection [106], [107], region growing [107], [108], etc. Yet, after many years of developments, unsupervised methods are still in need of higher accuracy and robustness to produce satisfying results. On the other hand, supervised methods primarily depend on their training datasets or user inputs as prior knowledge, the quality of which will directly affect the quality of their segmentation results. These methods are usually based on statistical models [109], [110], and population-based information is

represented from the training datasets. However, due to their dependency of prior knowledge, large-scale training data is usually required for higher performances, which is hard to obtain before the segmentation tasks.

Recently, the computer-aided semi-supervised segmentation methods have emerged as a popular compromise solution. The semi-supervised methods are generally able to provide more accurate and efficient segmentation results with minimum user inputs, hence have become the currently prevailing means of image segmentation. As a major branch of semi-supervised image segmentation methods, the graph-based methods exhibit remarkably elevated accuracy and robustness in comparison with other methods. They usually take advantage of user inputs to directly indicate clues about the foreground and background in the images. The segmentation problem is then solved by applying various graph theories. These methods are primarily variations of five graph theoretic techniques, namely graph cut, random walks, shortest path, power watershed, and minimum spanning tree:

(1) The graph cut method and its variations generally aim to solve energy minimization problems for low-level computer vision tasks. They can be reduced to instances of the max-flow/min-cut theorem [111]. Existing implementations include graph cut with cost functions [112], graph cut on Markov random filed (MRF) [113], and graph cut on conditional random field (CRF) [114], etc. As a major extension, the GrabCut method [115] applies user-specified bounding box with a Gaussian mixture model to estimate the color distribution of the object, and achieves relatively accurate results.

(2) The random walks method is initially introduced as a mathematical formalization of a random sequence path used in data classification [116]. It calculates the probability

from any element in an image to each of the user-defined seed points, and determines the cluster that an element most likely belongs to [117].

(3) The shortest path method aims to find a path between two vertices in a graph, in which the sum of weights of the edges passed is minimized. Existing methods include Dijkstra's method [118], Bellman-Ford method [119], A* search method [120], and Johnson's method [121], etc.

(4) The power watershed method inherits the basic ideas from the watershed method [122]. It is a generalized framework that effectively extends from graph cuts, random walks and shortest path methods. It is an integration of unary terms in a standard watershed method to improve the segmentation results.

(5) The minimum spanning tree method is a subcategory of the spanning tree method [123], which applies a tree structure that connects the vertices of a given graph together. There are multiple methods available, including Boruvka's method [124], Kruskal's method [125], Prim's method [126], and parallel method [127], etc.

For the graph-based semi-supervised methods above, user interactions are required. These methods model the image as a weighted graph to reflect local intensity changes, and a small number of user-provided seeds are applied to estimate the foreground and the background regions. The final solution is usually achieved by minimization of the corresponding energy function. For example, the graph cut method performs a max-flow/min-cut analysis to find the minimum weight cut between the seeds of the foreground and the seeds of the background. Another example is the random walks method, in which the diffusion distances are calculated as the classification probabilities [117], [128].

In this thesis, the ideas of weighted graph, random walks and energy function optimization are used to facilitate the saliency detection process, which will be discussed in Chapter 3 and Chapter 4.

2.3 Object Proposal Generation

Object proposal generation, or objectness measurement, is a class of methods that attempt to generate a small set (e.g. a few hundreds or thousands) of potential object regions (called object proposals) in a given image, so that these object proposals can cover the different objects in the image to the maximum extend, regardless of the specific categories of these objects (i.e. generic over categories) [129-134]. The object proposal generation is often adopted as a pre-processing stage before subsequent tasks. Compared with conventional sliding window based object detection paradigm [135], [136], object proposal generation has three major advantages:

(1) It better accords with human visual system which quickly perceives objects before identifying them [137], [138];

(2) It greatly speeds up the computation by reducing the potential candidates of search locations (e.g., from typically a few million candidates to less than a few thousand candidates), especially when the number of object classes that need to be detected is high；

(3) It also helps to improve the accuracy of the object detection task by allowing the usage of more powerful classifiers during testing, since it restricts the detection only on the object proposals [139].

Object proposal generation and saliency detection are closely correlated. On the one hand, the object proposal generation process consider saliency as a useful cue for

measuring objectness of a region [130], [140]; in other words, an object is more likely to be salient than a background region [141], [142]. On the other hand, the saliency detection process applies objectness measurements to distribute high and low saliency values to objects and background, which leads to higher accuracy [74].

In this thesis, the idea of object proposal is exploited to provide initial saliency estimations, which will be discussed in Chapter 5.

2.4 Deep Neural Network (DNN)

Deep neural network is a branch of machine learning that has experienced drastic developments in the last decade. First proposed by LeCun *et al.* in 1989 [143], the DNNs, and especially the convolutional neural networks (CNNs), are designed to model high-level nonlinear data features by multiple complex processing layers [59]. Since emergence, DNN has received remarkable success in image classification [144-146], object detection [147], [148], semantic segmentation [149-151], face recognition [152], [153], pose estimation [154], pedestrian behavior estimation [155], [156], and cancer type / subtype classification [157] etc.

The current applications of DNN focus on two major categories, namely sparse labeling and dense labeling. In this section, we will first briefly review the basic principles of neural networks, and then introduce the applications of DNN on sparse and dense labeling.

2.4.1 Fundamentals of Neural Networks

This section is based on the online course Unsupervised Feature Learning and Deep Learning [158]. Consider a supervised learning problem, where we have access to the

labeled training examples $(x^{(i)}, y^{(i)})$. The neural networks provide a way of defining a complex, non-linear model $h_{W,b}(x)$, with parameters W, b to be fitted by the training data. The simplest neural network, which consists of only a single "neuron", is shown in Figure 2.4.

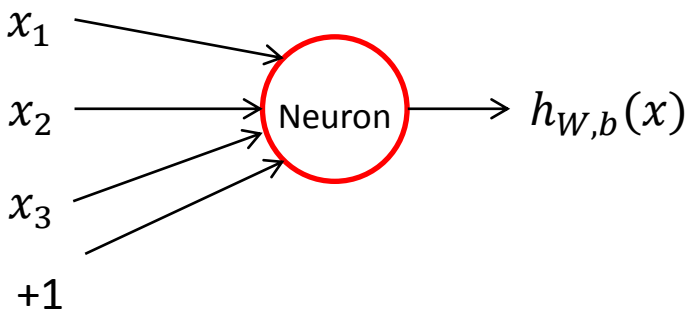

Figure 2.4 An illustrative diagram of a "neuron" in DNN. The X1~X3 stand for inputs, and "+1" stands for bias.

A neural network is usually established by hooking together many of the "neuron" structures in Figure 2.4, so that the output of one neuron is the input of another. For instance, Figure 2.5 shows a typical neural network with 3 layers. In this structure, the "+1" are bias units; the leftmost layer is the input layer, usually takes in images or other data structures; the rightmost layer is the output layer, which can output a single label (for sparse labeling) or a matrix-like label mask (for dense labeling); the middle layer is a hidden layer, since its values are not directly observed during the training process.

The network has parameters $(W, b) = (W^{(1)}, b^{(1)}, W^{(2)}, b^{(2)})$, where $W_{ij}^{(l)}$ and $b_{ij}^{(l)}$ denote the weight and bias associated with layer l. The $a_i^{(l)}$ denote the activation value of unit i in layer l. For $l = 1$ (the input layer), $a_i^{(1)} = x_i$ is defined. The computation that the network represents is given by the equations below, which are called forward propagation; the network is hence called feed-forward network:

$$
\begin{aligned}
a_1^{(2)} &= f(W_{11}^{(1)}x_1 + W_{12}^{(1)}x_2 + W_{13}^{(1)}x_3 + b_1^{(1)}) \\
a_2^{(2)} &= f(W_{21}^{(1)}x_1 + W_{22}^{(1)}x_2 + W_{23}^{(1)}x_3 + b_2^{(1)}) \\
a_3^{(2)} &= f(W_{31}^{(1)}x_1 + W_{32}^{(1)}x_2 + W_{33}^{(1)}x_3 + b_3^{(1)}) \\
h_{W,b}(x) = a_1^{(3)} &= f(W_{11}^{(2)}a_1^{(2)} + W_{12}^{(2)}a_2^{(2)} + W_{13}^{(2)}a_3^{(2)} + b_1^{(2)})
\end{aligned}
\tag{2.1}
$$

After the computation hits the output layer, there will be a cost function $J(W,b)$ calculating the cost against the ground truth label. The cost is then propagated backwards with gradients of each layer to update the parameters W,b, which is called backpropagation:

$$
\begin{aligned}
W_{ij}^{(l)} &= W_{ij}^{(l)} - \alpha \frac{\partial}{\partial W_{ij}^{(l)}} J(W,b) \\
b_i^{(l)} &= b_i^{(l)} - \alpha \frac{\partial}{\partial b_i^{(l)}} J(W,b)
\end{aligned}
\tag{2.2}
$$

During training, the forward propagation and backpropagation are conducted alternately to update the network parameters, until the cost is small enough, or the maximum iteration number is reached.

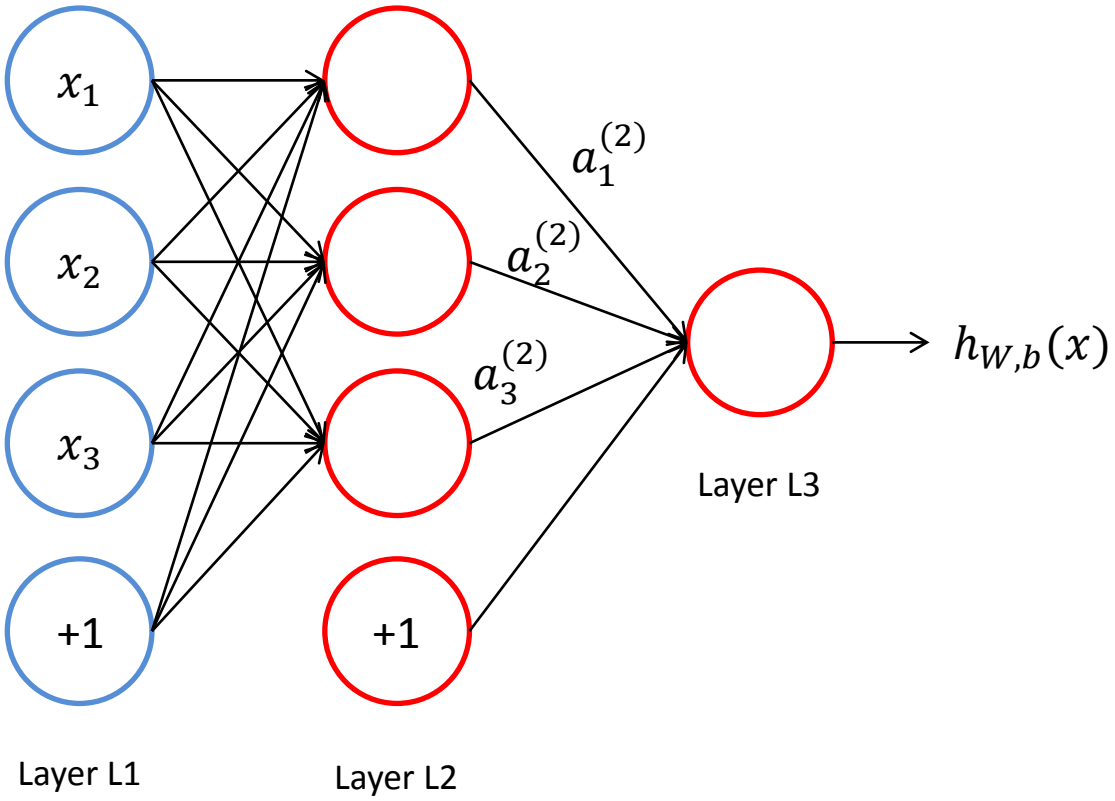

Figure 2.5 A small illustrative neural network with 3 layers. The X1~X3 stand for inputs, and "+1" stands for bias.

In practice, instead of the fully connected network in Figure 2.5, the convolutional neural network (CNN) is more prevalently used. CNN is also a type of feed-forward neural network, which models the animal visual perception. Instead of full connection, the connection between different layers of CNN is realized by 2D-convolution. Figure 2.6 exhibits a typical architecture of CNN.

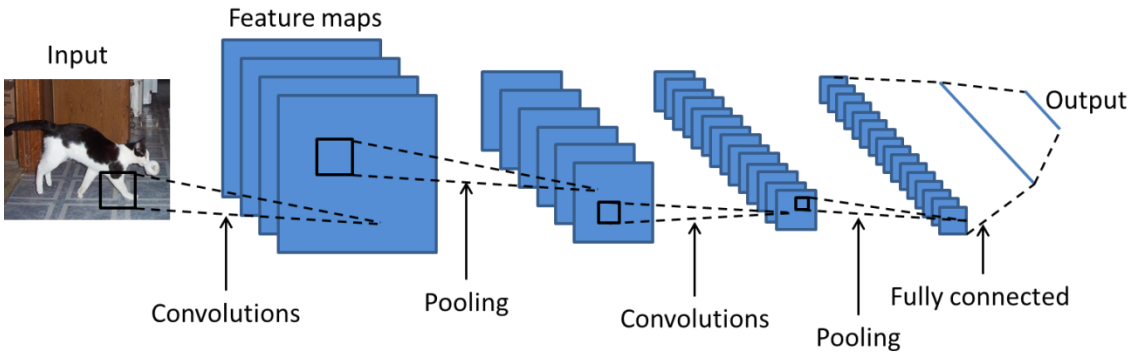

Figure 2.6 A typical CNN architecture.

Compared with fully connected network, CNN benefits from the shared weights of each convolution filter among layers, which greatly reduce the overall size of the network. CNN is also renowned for being space invariant, which contributes to its significantly elevated robustness. Given these advantages, CNN receives broad applications in various computer vision tasks.

In this thesis, CNN is applied in all of our DNN models, which will be introduced in Chapter 5.

2.4.2 DNN Based Sparse Labeling

Sparse labeling is the fundamental application of DNN in classification tasks. The idea is to generate a single class label for each input sample [159], such as an image. Many state-of-the-art network models are designed under this scheme, including AlexNet [144], OverFeat [145], Clarifai [160], VGG [161], GoogLeNet [146], and ResNet [162], etc. Recently, initial studies have emerged towards the application of DNN on saliency sparse labeling. For instance, Wang *et al.* [90] train two separate DNNs with image patches and object proposals for local and global saliency; Zhao *et al.*[91] establish a multi-context DNN model for superpixel-wise saliency classification; and Li *et al.* [92]

propose a multi-scale DNN model for feature extraction, the outputs of which are then aggregated for the final saliency map.

2.4.3 DNN Based Dense Labeling

On the other hand, the dense labeling is a newly arising application of DNN that has drawn much attention. Unlike sparse labeling, dense labeling aims to predict a complete label mask (instead of a single label) based on the input sample, with either identical or reduced size. Since much more per-sample label information can be generated than sparse labeling, DNN-based dense labeling has greatly facilitated many previously challenging tasks such as object detection and semantic segmentation, in terms of both accuracy and efficiency. In [147], Szegedy *et al.* propose the idea of DNN-based object detection via DNN regression and multi-scale refinements. Girshick *et al.*[148] combine CNNs with bottom-up region proposals to localize and segment objects. Long *et al.*[150] propose the idea of fully convolutional network (FCN), which achieves dramatic improvements in semantic segmentation. And in the work of Chen *et al.*[151], responses from CNNs are combined with fully connected CRF, which overcomes the poor localization property of CNN itself.

There have been multiple methods in exploring for the application of DNN on saliency detection, such as [90-93]. The details about how DNN greatly facilitated the performance of our proposed DSL method will be discussed in Chapter 5. We will also see that in general, the DNN-based methods greatly outperforms conventional low-level feature based methods, which is attributed to their learning processes.

Chapter 3 Conventional Low-Level Feature Based Saliency Detection

In the first two chapters of this thesis, the general background of saliency detection is introduced, which covers the origins of saliency detection, the challenges faced in existing methods, and the related works in various aspects. Starts from Chapter 3, we will present our proposed saliency detection methods in detail.

In this chapter, we propose a novel conventional low-level feature based saliency detection method, the regularized random walks ranking (RR) method. Section 3.1 summarizes the challenge we are going to address; section 3.2 lists our contributions and the two major steps in our proposed RR method; section 3.3 reviews the models of manifold ranking and random walks as related works; section 3.4 illustrates the methodology step by step; section 3.5 includes experimental results and discussion; and finally, section 3.6 concludes this chapter.

3.1 Problem Formulation

As introduced in Chapter 1.3.1, in the field of saliency detection, many graph-based algorithms heavily depend on the accuracy of the pre-processed superpixel segmentation, which leads to significant sacrifice of detail information from the input image. On the other hand, a part of existing methods are based on problematic pre-assumptions to guide the saliency estimation, which are easily violated on broader datasets with more unusual-patterned images. As a typical example, the MR method [52] adopts the four boundaries of the input image as background reference, which is implausible in many

cases. In other words, one or more boundaries may be adjacent to the foreground object and undesirable results may emerge if we still use them as background queries, as shown in Figure 1.2. Another drawback of MR is that it depends on the pre-processed superpixel segmentation, whose inaccuracy may directly lead to the failure of the entire algorithm. Besides, assigning the same saliency value to all pixels in a superpixel node cannot exploit the full potential of the detail information from the original image.

3.2 Contributions

To address the issues above, we propose a novel bottom-up saliency detection method that takes the advantage of both region-based features and image details. Our method has two main steps:

(1) We first optimize the image boundary selection by the proposed erroneous boundary removal step.

(2) By taking the image details and region-based estimations into account, we then propose the regularized random walks ranking (RR) to formulate pixel-wised saliency maps from the superpixel-based background and foreground saliency estimations.

Experimental results on two public datasets indicate the significantly improved accuracy and robustness of our proposed algorithm, in comparison with 12 state-of-the-art saliency detection methods.

3.3 Related Works

In this section, as preliminary knowledge introduction, we provide a brief review of the manifold ranking model and the random walks model, which are closely related to our proposed RR method.

3.3.1 Manifold Ranking

Manifold ranking is a kind of ranking algorithm that is initially used in pattern classification [163], [164]. It assigns ranks to the elements in a dataset, which reveal their likelihood being in a certain class with respect to their intrinsic manifold structure.

Given a dataset $\chi = \{x_1, ..., x_s, x_{s+1}, ..., x_n\} \in \mathbb{R}^m$, where n is the element number, the first s elements are the labeled queries, while the rest are the unknown elements that need to be ranked. This identification is recorded in an indication vector $y = [y_1, ..., y_n]^T$, where $y_i = 1$ if x_i belongs to the queries, and $y_i = 0$ otherwise. Note that if prior knowledge about the confidences of the queries is available, we can assign different ranking scores to the queries proportional to their confidences, instead of just 0 and 1.

The manifold algorithm functions in the following steps:

1) Sort the pairwise distance of elements in ascending order. Repeat connecting two elements with an edge according the order until a fully connected graph is formed.

2) Establish the weight matrix $W = [w_{ij}]_{n \times n}$ linking x_i and x_j. Note that $W_{ii} = 0$.

3) Symmetrically normalize W by $S = D^{-1/2} W D^{-1/2}$, where $D = diag(d_1, ..., d_n)$ is the degree matrix with

$$d_i = \sum_j w_{ij} \tag{3.1}$$

When applied to graphs, a graph structure $G = (V, E)$ with nodes V and edges E is first established, where V corresponds to the dataset χ, and E collects all the connections of any two nodes in G quantified by the weight matrix W.

Let $f : \chi \to \mathbb{R}^n$ be the ranking function assigning rank values $f = [f_1, ..., f_n]^T$ to χ, which would be obtained by solving the following minimization problem,

$$f^* = \arg\min_f \frac{1}{2}\left(\sum_{i,j=1}^{n} w_{ij} \left\| \frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right\|^2 + \mu \sum_{i=1}^{n} \| f_i - y_i \|^2 \right) \tag{3.2}$$

where μ is a controlling parameter. The optimized solution is given in [52], [164], [165] as

$$f^* = (D - \alpha W)^{-1} y \tag{3.3}$$

where $\alpha = 1/(1+\mu)$.

The manifold ranking model is used to estimate the rough saliency in our proposed RR method, which will be presented in section 3.4. The input image is first segmented into n superpixels via the simple linear iterative clustering (SLIC) approach [166]. A superpixel-based graph $G = (V, E)$ is subsequently constructed with nodes V as superpixels. The edge set E is defined with the following three criteria [52]:

1) Neighboring nodes with shared edges are connected to each other;

2) Each node is also connected to the neighbor nodes of its own neighbors;

3) Any two nodes from the four boundaries of the graph are treated as connected.

The weight matrix W is established based on E, in which the weight of adjacent nodes is defined as

$$w_{ij} = \exp\left(-\frac{\| c_i - c_j \|^2}{\sigma^2} \right) \tag{3.4}$$

where c_i and c_j are the mean CIELab colors of the two nodes i and j, and σ is a controlling constant. The remaining elements of W for the unconnected nodes are all assigned as zeros, and the degree matrix D is computed in (3.1).

3.3.2 Random Walks

Random walks is a mathematical formalization of a random sequence path, which leads an element to a seed location with the highest likelihood [117]. Given a dataset $\chi = \{x_1, ..., x_n\} \in \mathbb{R}^m$, where n is the element number, the task is to group the elements into K classes. We first mark s elements from χ as the seed nodes with at least one element of each class. Without loss of generality, we assume that the first s elements of χ are the seeds, so that $\chi = [x_M^T, x_U^T]$, in which x_M are the seed nodes and x_U are the unseeded nodes. The graph $G = (V, E)$, weight matrix W, and degree matrix D are constructed similarly to those in section 3.3.1. We further define the $n \times n$ Laplacian matrix L as

$$L_{uv} = \begin{cases} d_u & \text{if } u = v, \\ -w_{uv} & \text{if } x_u \text{ and } x_v \text{ are adjacent nodes,} \\ 0 & \text{otherwise.} \end{cases} \tag{3.5}$$

Note that we use u and v as element subscripts in pixel-wise graphs to differentiate from i and j used in superpixel-wise graphs. Since the edges E are undirected, L is symmetric. Accordingly, we define the label function for seed nodes as

$$Q(x_u) = k, k \in \mathbb{Z}, 0 < k \le K \tag{3.6}$$

Then we let $p^k = \left[p_1^k, \ldots, p_n^k \right]^T$ denote the probability vector of χ for label k, which can similarly be partitioned as $p^k = \left[\left(p_M^k \right)^T, \left(p_U^k \right)^T \right]$. Here p_M^k is for the seed nodes, which has fixed value as

$$p_u^k = \begin{cases} 1 & Q(x_u) = k, \\ 0 & \text{otherwise.} \end{cases} \tag{3.7}$$

The optimized p^k is achieved by minimizing the Dirichlet integral [117], [128],

$$\begin{aligned} Dir\left[p^k\right] &= \frac{1}{2}\left(p^k\right)^T L\left(p^k\right) \\ &= \frac{1}{2}\left[\left(p_M^k\right)^T \left(p_U^k\right)^T\right]\begin{bmatrix} L_M & B \\ B^T & L_U \end{bmatrix}\begin{bmatrix} p_M^k \\ p_U^k \end{bmatrix} \end{aligned} \tag{3.8}$$

We differentiate $Dir[p^k]$ with respect to p_U^k, and the critical point is found as

$$p_U^k = -L_U^{-1} B^T p_M^k \tag{3.9}$$

In section 3.4.3, the random walks model is reformulated for the final saliency map computation. The graph $G = (V, E)$ is pixel-wise, and the weight matrix W is defined as

$$w_{uv} = \exp\left(-\frac{\left\|g_u - g_v\right\|^2}{\sigma^2}\right) \tag{3.10}$$

where g_u and g_v are the intensities at pixel u and v, and σ is the same controlling constant used in (3.4).

3.4 Saliency Detection with Regularized Random Walks Ranking (RR)

The proposed RR method consists of three major steps. Step one removes the boundary with the lowest probability belonging to the background, and generates saliency estimation via the refined background queries; step two generates foreground saliency estimation based on the complementary values of the background estimation; step three extracts seed references from step two, and calculates the pixel-wise saliency map with the proposed regularized random walks ranking.

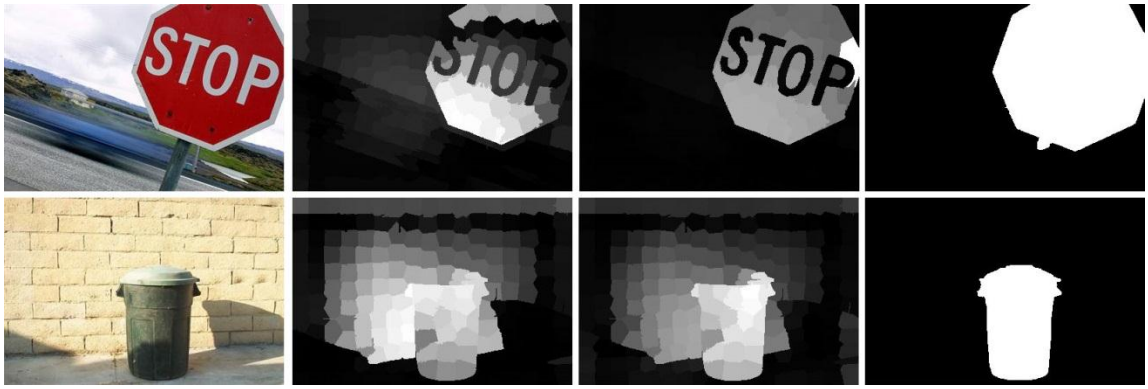

Figure 3.1 The effect of erroneous boundary removal in section 3.4.1. From left to right: input images, background saliency estimations with all boundaries, background saliency estimations after erroneous boundary removal, ground truth.

3.4.1 Background Saliency Estimation

As stated in section 1.3.1, it is possible for a boundary in the input image to be occupied by the foreground object. Using such a problematic boundary as queries in the background saliency estimation may lead to undesirable results, such as the typical example illustrated in the second column of Figure 3.1. We therefore optimize the boundary queries by locating and eliminating the erroneous boundaries before the background saliency estimation.

Given the conspicuous difference of color and contrast between the background and the salient object, the erroneous boundary tends to have distinctive color distribution compared to the remaining three. Hence, we treat the superpixel boundaries as connected regions, and calculate their normalized pixel-wise RGB histogram respectively,

$$H_b(h) = \frac{1}{l}\sum_{q=1}^{l}\delta\left(I_q - h\right) \tag{3.11}$$

where $b \in \{top, bottom, left, right\}$ indicates the four boundary locations; l is the total pixel number in the target region; $h = 0,...,255$ is the intensity bin variable; I_q is the

intensity value of pixel q; and $\delta(\cdot)$ is the unit impulse function. The red, green and blue channels are calculated separately using 256 bins. We then compute the Euclidean distance of any two of the four histograms,

$$A(b_1,b_2)=\sqrt{\sum_{h=0}^{255}\left[\left(H_{b_1}^{red}(h)-H_{b_2}^{red}(h)\right)^2+\left(H_{b_1}^{green}(h)-H_{b_2}^{green}(h)\right)^2+\left(H_{b_1}^{blue}(h)-H_{b_2}^{blue}(h)\right)^2\right]} \quad (3.12)$$

This results in a 4×4 matrix A, which is then summed in column-wise. The maximum of the summation determines the boundary to be removed. E.g. if the second column sums to be the largest, the bottom boundary will be removed.

The superpixels on each of the three remaining sides of the image will be labeled as ones in the indication vector y in (3.2), while other nodes as zeros. Three ranking results f_l^* will be achieved afterwards based on (3.3), where l corresponds to the three remaining locations. Since the ranking results show the background relevance of each node, we still need to calculate their complement values to obtain the foreground-based saliency,

$$S_l(i)=1-f_l^*(i),\ i=1,...,n \quad (3.13)$$

where n is the total superpixel number. The results are then put into element-by-element multiplication to calculate the saliency estimation result of this section,

$$S_{step1}(i)=\prod_l S_l(i). \quad (3.14)$$

The major advantage of erroneous boundary removal is that it helps to relieve the inaccuracy of using all boundaries in cases that one or more of the boundaries happen to be adjacent to the foreground object. As shown in Figure 3.1, removal of the most irrelevant boundary (right for the first row, and bottom for the second row) leads to more accurate outputs.

3.4.2 Foreground Saliency Estimation

Section 3.4.1 calculates the foreground saliency by complementary subtraction of the background saliency estimation, which leads to favorable results in images with conspicuous contrasts between the foreground and the background. However, the background queries alone are sometimes insufficient to fully illustrate the foreground information, especially in cases where the salient object has complicated structure or similar patterns to the background. Subsequent foreground-query-based saliency estimation is hence desired.

The foreground queries are obtained by extracting S_{step1} with a threshold $t = mean(S_{step1})$, followed by re-performance of (3.3) with the newly defined indication vector y. The ranking function f can be directly calculated from (3.3) and is treated as the foreground saliency estimation as follows,

$$S_{step2}(i) = f(i), i = 1, ..., n \tag{3.15}$$

which will be used in the next step as seed references.

3.4.3 Saliency Map Formulation by Regularized Random Walks Ranking

Former manifold-ranking-based saliency detection [52] completely depends on the SLIC superpixel segmentation, which may generate undesirable results if the superpixel segmentation itself is imprecise. In addition, assigning the same saliency value to all pixels within a same node enormously sacrifices the detail information. To overcome these disadvantages, we develop a regularized random walks ranking model to formulate

saliency maps, which is independent of the superpixel segmentation, and may reveal pixel-wised saliency map of the input image.

The regularized random walks ranking is extended from the random walks model introduced in section 3.3.2. We suggest a fitting constraint, which restricts the Dirichlet integral to be as close to the prior saliency distribution as possible,

$$Dir\left[p^k\right]=\frac{1}{2}\left(p^k\right)^T L\left(p^k\right)+\frac{\mu}{2}(p^k-Y)^T(p^k-Y) \tag{3.16}$$

where μ is the same controlling parameter used in (3.2), and Y is a pixel-wise indication vector inheriting the values of S_{step2}. In other words, different pixels within a same superpixel in S_{step2} share the same saliency value in Y. Note that the regularized random walks ranking is computed in pixel-wise, thus both p^k and Y are $N\times 1$ vectors, and L is an $N\times N$ matrix, where N is the total pixel number in the image. We define two thresholds t_{high} and t_{low} as follows,

$$\begin{aligned} t_{high} &= \frac{\text{mean}(S_{step2})+\max(S_{step2})}{2} \\ t_{low} &= \text{mean}(S_{step2}), \end{aligned} \tag{3.17}$$

which are used to select pixels with $Y_u > t_{high}$ as foreground seeds, and $Y_u < t_{low}$ as background seeds. The seeds are then combined into $p_M^k, k=1,2,\ldots$, where $k=1$ corresponds to the background label, and $k=2$ corresponds to the foreground label. The matrix decomposition of (3.16) is conducted as follows,

$$\begin{aligned} Dir\left[p^k\right] &= \frac{1}{2}\left[\left(p_M^k\right)^T\left(p_U^k\right)^T\right]\begin{bmatrix} L_M & B \\ B^T & L_U \end{bmatrix}\begin{bmatrix} p_M^k \\ p_U^k \end{bmatrix} \\ &+\frac{\mu}{2}\left(\begin{bmatrix} p_M^k \\ p_U^k \end{bmatrix}-\begin{bmatrix} Y_M^k \\ Y_U^k \end{bmatrix}\right)^T\left(\begin{bmatrix} p_M^k \\ p_U^k \end{bmatrix}-\begin{bmatrix} Y_M^k \\ Y_U^k \end{bmatrix}\right) \end{aligned} \tag{3.18}$$

After setting the differentiation of $Dir\left[p^k\right]$ with respect to p_U^k as zero, the optimized solution is obtained,

$$p_U^k = \left(L_U + \mu I\right)^{-1}\left(-B^T p_M^k + \mu Y_U^k\right) \tag{3.19}$$

p_U^k and p_M^k are then combined to form p^k. We set $k=2$ to select the foreground possibility p^2 and reshape it to a matrix S_{final} with same size of the input image as the final foreground saliency output.

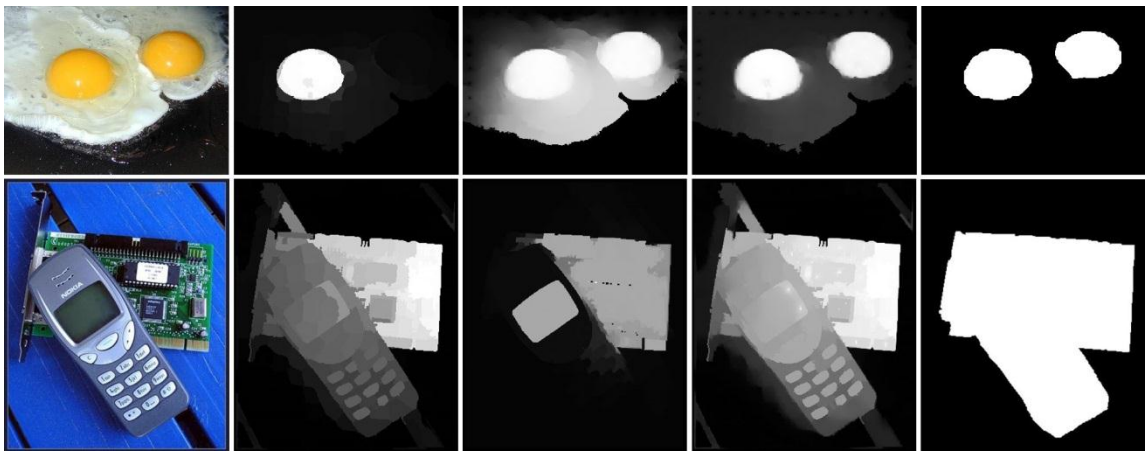

Figure 3.2 Examples that (3.16) leads to more precise saliency outputs. From left to right: input images, saliency estimation results, saliency outputs with random walks, saliency outputs with regularized random walks ranking, ground truth.

Since the seeds are automatically generated from the result of section 3.4.2, unlike classical random walks [117], no user interaction is required. The fitting constraint in (3.16) provides a prior saliency estimation to all pixels instead of the seed pixels alone, which offers a better guidance in calculating the final saliency map. The effect of the fitting constraint in (3.16) is shown in Figure 3.2, where the regularized random walks ranking not only greatly improves the saliency map from the previous saliency estimation step, but also remarkably outperforms random walks.

The main process of our proposed algorithm is summarized in Table 3.1.

Table 3.1 Algorithm description of our proposed RR method

Step	Content
Input	An image and related parameters.
1	Establish the graph structure with superpixels as nodes; calculate W and D with (3.4) and (3.1).
2	Conduct erroneous boundary removal with (3.12).
3	Acquire the background saliency estimation S_{step1} with (3.14).
4	Acquire the foreground saliency estimation S_{step2} with (3.15).
5	Establish the pixel-wise graph structure and obtain L with (3.5); then compute the saliency possibilities p^k with (3.19).
6	Set $k = 2$ and reshape p^2 into S_{final} as the final saliency output.
Output	A saliency map with the same size as the input image.

3.5 Experimental Results

In this section, we present the experimental results of our proposed RR method. We first introduce the datasets, evaluation metrics and algorithm parameters we used, then evaluate the two design options in our method (i.e. erroneous boundary removal, and regularized random walks ranking), and finally exhibit the comparison experiment against 12 state-of-the-art saliency detection methods. The efficiency and limitation of our method are also presented.

3.5.1 Datasets

Two public datasets are adopted in our experiments:

(1) The MSRA10K dataset [63], [64], which contains 10,000 randomly chosen images from the MSRA dataset [8], [167];

(2) The DUT-OMRON dataset [52], which contains 5,168 manually selected highly-complex images.

Both datasets come with human-labeled ground truth. In our evaluation, we use all of the images in the datasets.

3.5.2 Evaluation Metrics

In referring to the experimental evaluations of most existing saliency detection methods, we use precision, recall and F-measure as our evaluation metrics. These terms are defined in [168] as,

$$\begin{aligned} precision &= \frac{\sum_{i=1}^{N} G(i) \cdot S_{final}(i)}{\sum_{i=1}^{N} S_{final}(i)}, \\ recall &= \frac{\sum_{i=1}^{N} G(i) \cdot S_{final}(i)}{\sum_{i=1}^{N} G(i)}, \\ F_{\beta} &= \frac{(1+\beta^2)\, precision \cdot recall}{\beta^2\, precision + recall} \end{aligned} \tag{3.20}$$

where $G(i)$ is the corresponding pixel-wise ground truth. In other words, precision is the ratio of retrieved true salient pixels to all the salient pixels retrieved, and recall is the ratio of retrieved true salient pixels to all the true salient pixels in the image.

Since the two terms precision and recall are in general contradictive to each other, i.e. the unilateral promotion of one term will often result in the deterioration of the other, the F-measure is prevalently adopted as a weighted average between precision and recall. We set $\beta^2 = 0.3$ to grant more importance to the precision, as suggested in [66].

In practice, precision and recall are usually displayed pairwise as the precision-recall (PR) curves, which are constructed by binarizing the saliency map with thresholds from 0 to 255.

3.5.3 Parameters

To conduct fair experimental comparisons, we adopt the same parameter settings in [52], where the superpixel number is set to $n = 200$, and the two controlling parameters are set to $\sigma^2 = 0.1$ and $\mu = 0.01$, respectively. No particular parameter needs to be defined in the proposed regularized random walks ranking algorithm.

3.5.4 Implementation

Our experiments are conducted in MATLAB on a 64-bit PC with Intel Core i5-4570 CPU @ 3.2 GHz and 8GB RAM. The MATLAB implementation of the proposed method is available at our website: https://github.com/yuanyc06/rr/.

3.5.5 Evaluation of Design Options

We first examine the major innovations of our proposed algorithm on the MSRA10K dataset, as shown in Figure 3.3. The blue and green curves illustrate the final saliency output comparison with and without the erroneous boundary removal. Obviously the erroneous boundary removal promotes the curve of the proposed method to a higher level. After that, we generate the saliency maps right after section 3.4.2 without using regularized random walks ranking. As shown by the blue and red curves in Figure 3.3, the complete algorithm also excels the algorithm without using regularized random walks ranking.

Based on the observations above, both the erroneous boundary removal and the regularized random walks ranking have contributions to the overall performance. We therefore adopt both of them in the following evaluations.

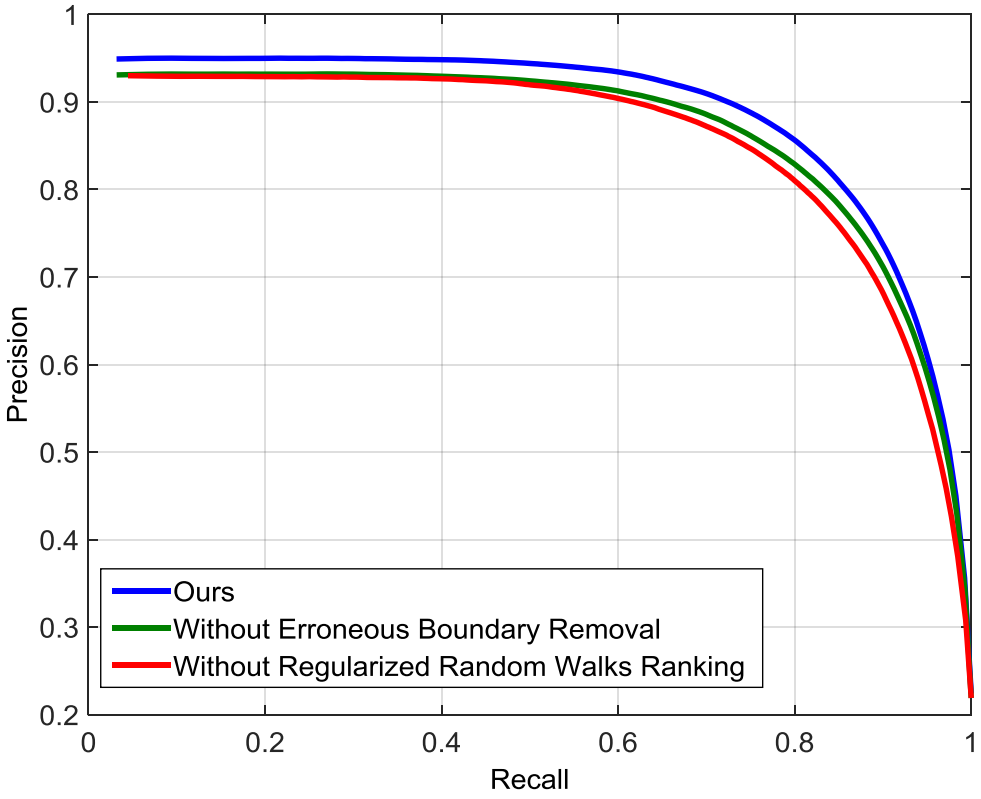

Figure 3.3 Precision-recall curves on the MSRA10K dataset with different design options of the proposed approach.

3.5.6 Evaluation Against State-of-the-Art

We then evaluate our proposed algorithm against twelve state-of-the-art saliency detection approaches, namely CA [78], CB [75], FT [66], GS [71], IT [5], LR [77], MR [52], PBO [169], PCA [79], SEG [61], SF [70] and SR [67].

The evaluation is first performed on the MSRA10K dataset, the results of which are shown in Figure 3.4 to Figure 3.6. The precision-recall curves in Figure 3.4 and Figure 3.5 demonstrate that the proposed method obviously outperforms all of the state-of-the-art algorithms. The proposed method is especially better than CA and CB, which are two of the top-performance algorithm from a recent benchmark of saliency detection [8]; the proposed method also completely excels its predecessor, i.e. the MR method, which

embodies the integrated strength of the improvements we made. On the other hand, Figure 3.6 demonstrates the F-measure comparison; the proposed method achieves the highest F-measure score 0.855, which is 1.06% over the second best algorithm (MR, 0.846).

To provide a qualitative comparison of the different saliency outputs, we select five example saliency maps from each of the thirteen methods, and tile them in Figure 3.7. The methods are sorted by the F-measure in Figure 3.6. We notice that our proposed RR method generates saliency maps with clearer details and finer boundary adherences.

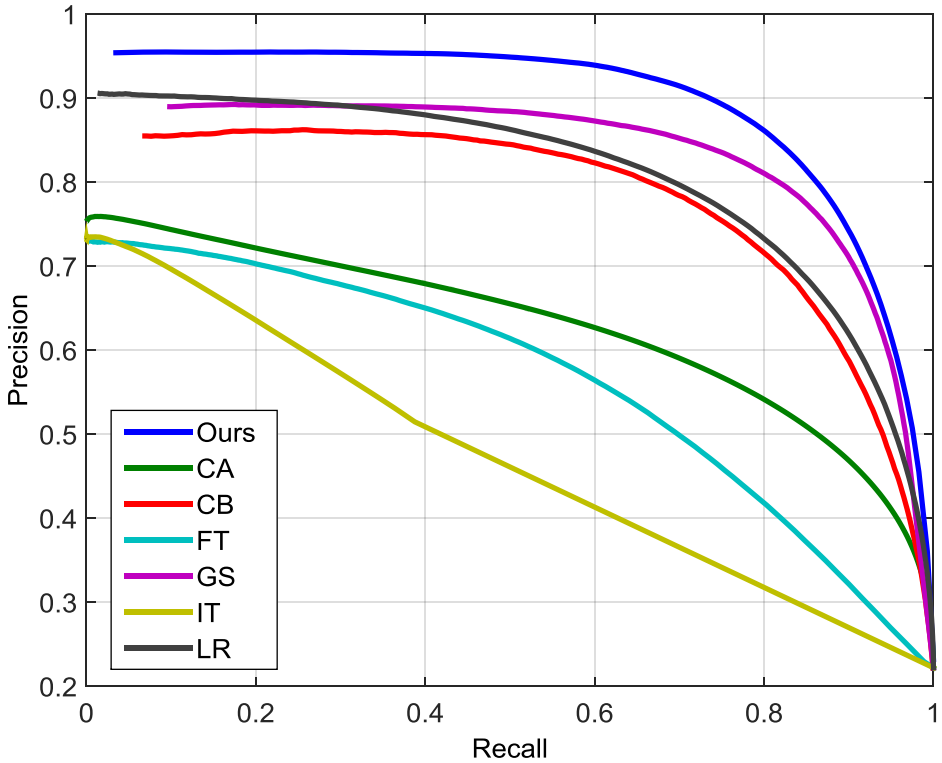

Figure 3.4 Precision-recall curves (part 1) of different methods on the MSRA10K dataset.

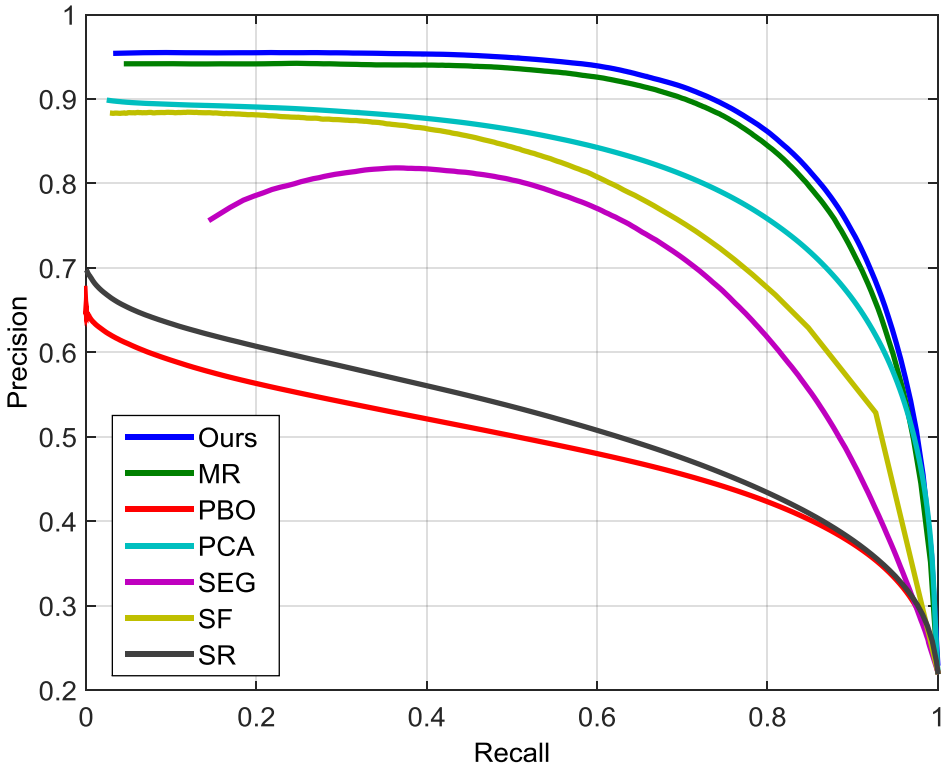

Figure 3.5 Precision-recall curves (part 2) of different methods on the MSRA10K dataset.

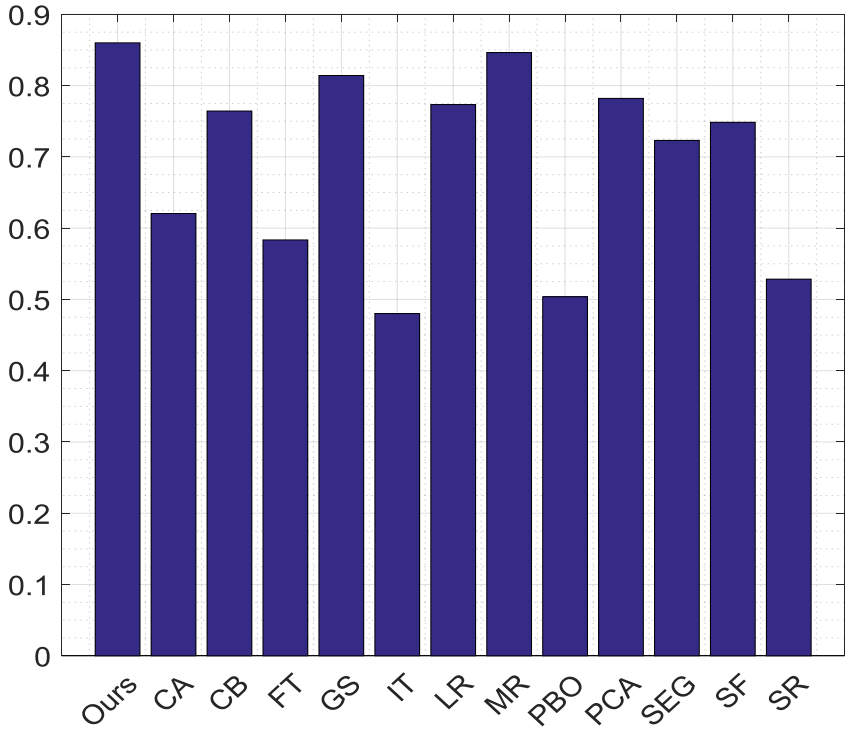

Figure 3.6 Average F-measures of different methods on the MSRA10K dataset.

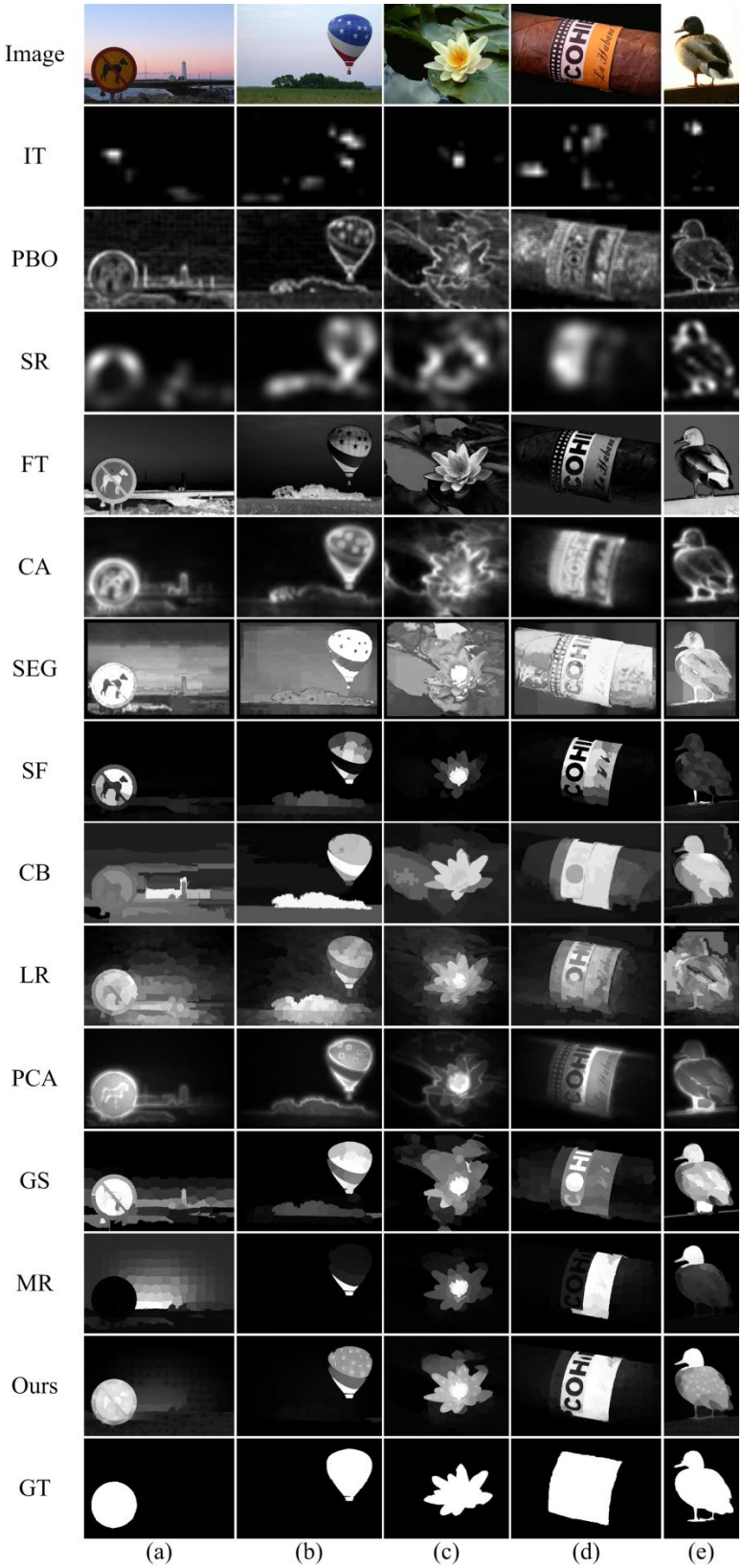

Figure 3.7 Saliency map examples of different methods on the MSRA10K dataset. (a) – (e): Image case IDs.

Next, we further evaluate the proposed algorithm on the DUT-OMRON dataset. The experiment process and evaluation metrics are the same as what we applied on the MSRA10K dataset. Precision-recall curves are shown in Figure 3.8 and Figure 3.9, and the F-measure comparison is shown in Figure 3.10. Again, our method outperforms all of the other approaches throughout different precision-recall curves. It also has the optimal F-measure 0.615, which is 0.82% over the second best algorithm (MR, 0.610). Besides the comparison among algorithms, we also notice that the performance of all methods on the DUT-OMRON dataset is in general far poorer than those on the MSRA10K dataset, which indicates that the images in DUT-OMRON are more challenging than MSRA10K, and higher performance on more challenging datasets is one of the potential directions of improvement to the proposed method.

Similar to Figure 3.7, we also select five example saliency maps from each of the thirteen methods, and tile them in Figure 3.11. Again, our proposed RR method outperforms the comparison method on various challenging cases, such as the images with boundary-adjacent salient objects (Figure 3.7(a) – (d)), or images with low contrast (Figure 3.7(c) and Figure 3.7(e)).

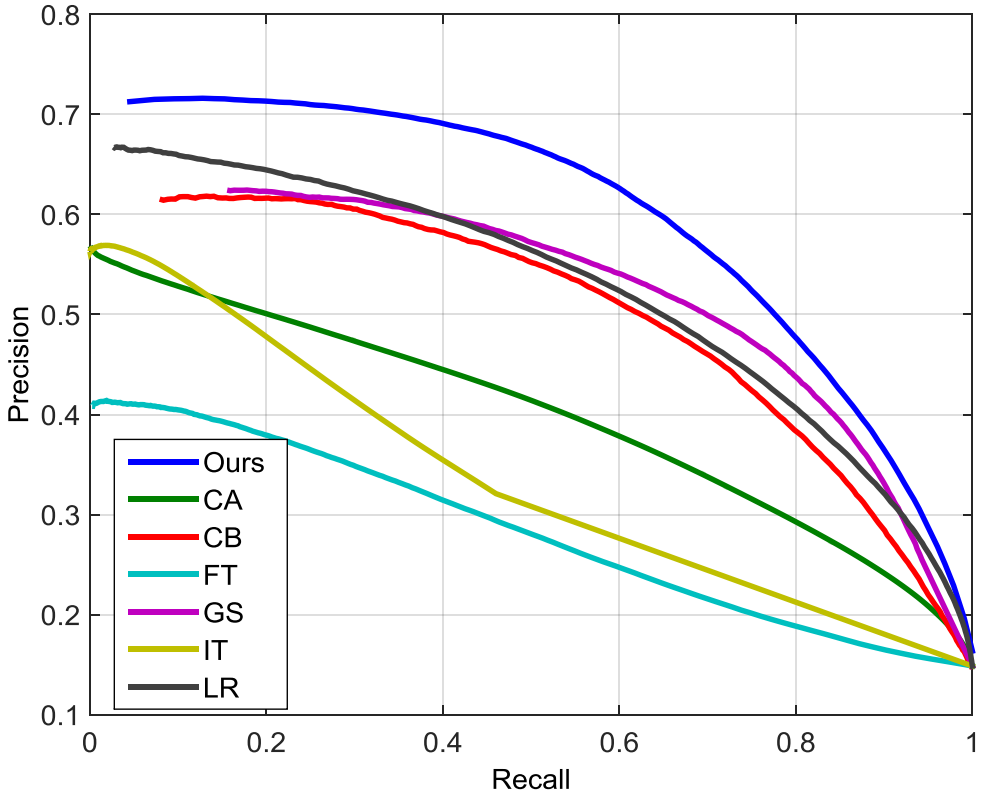

Figure 3.8 Precision-recall curves (part 1) of different methods on the DUT-OMRON dataset.

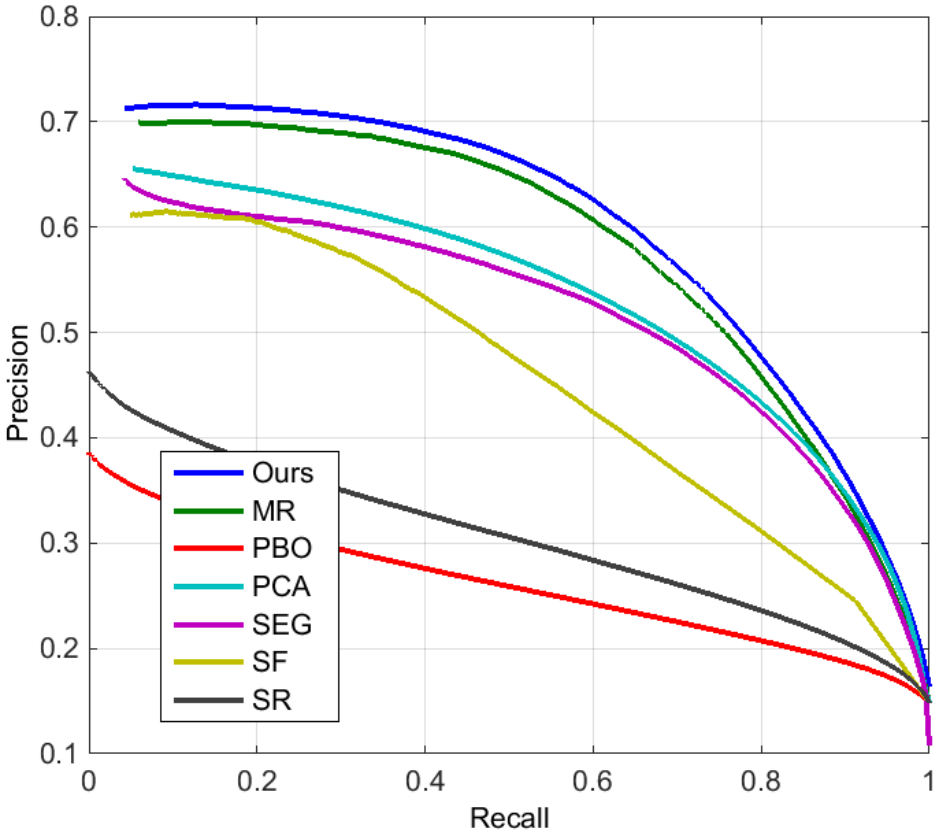

Figure 3.9 Precision-recall curves (part 2) of different methods on the DUT-OMRON dataset.

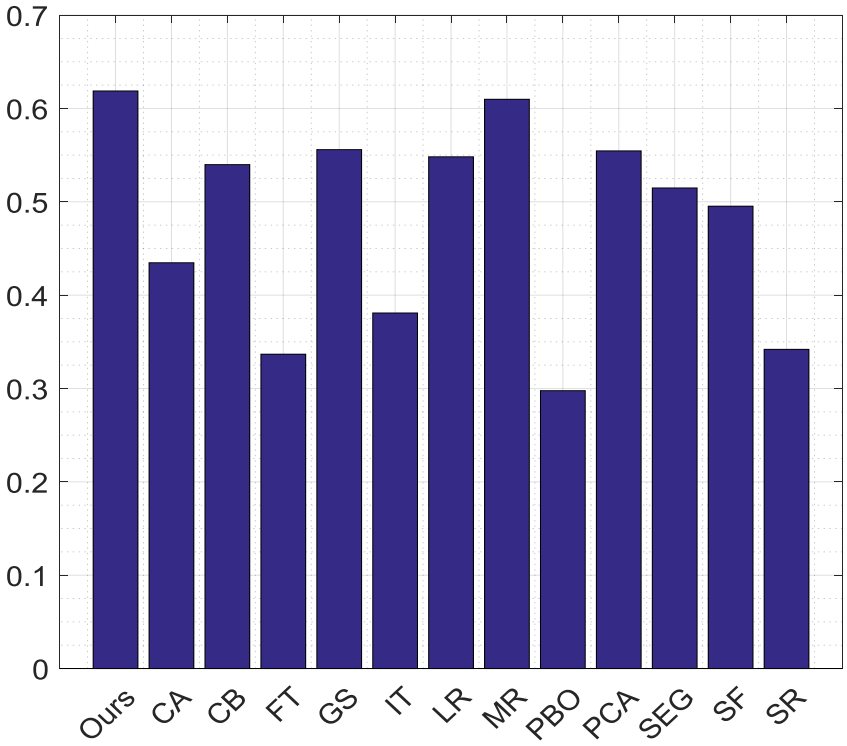

Figure 3.10 Average F-measures of different methods on the DUT-OMRON dataset.

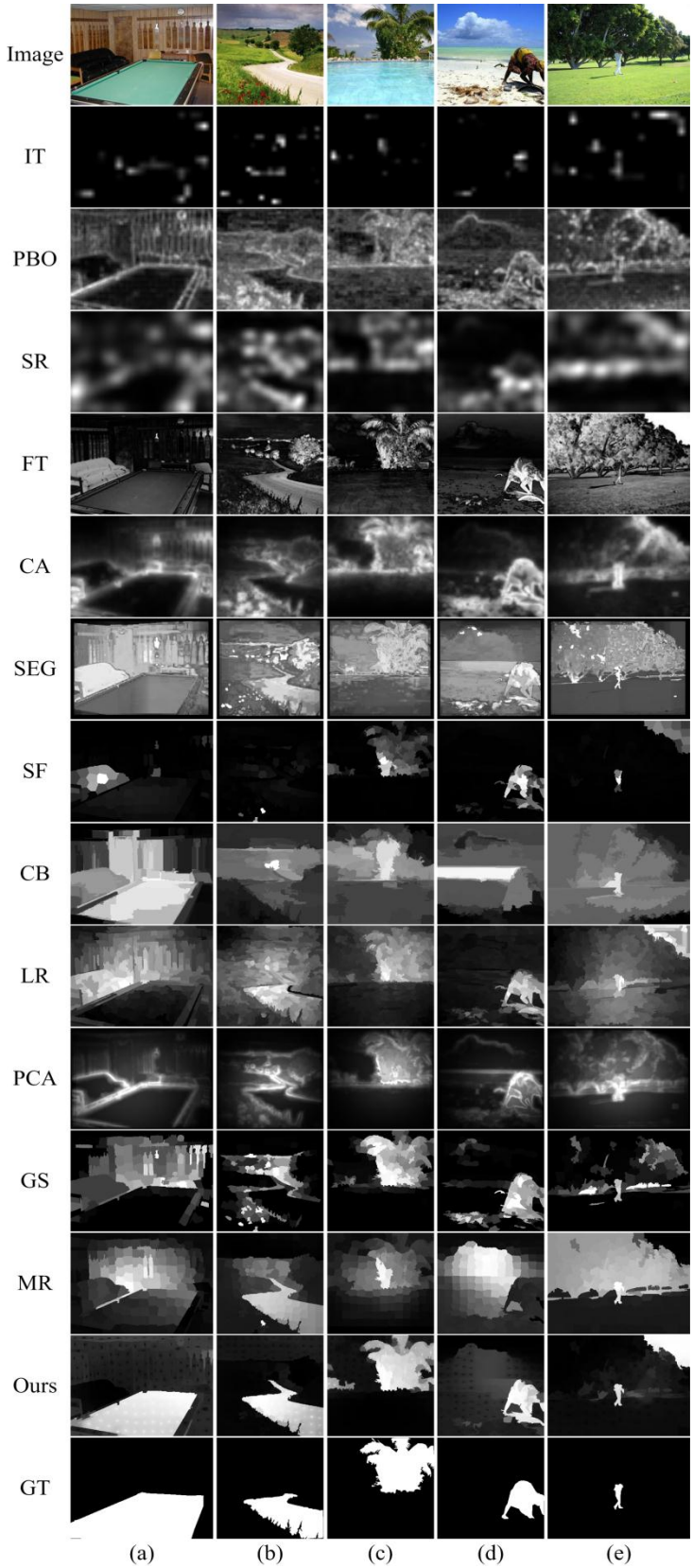

Figure 3.11 Saliency map examples of different methods on the DUT-OMRON dataset. (a) – (e): Image case IDs.

3.5.7 Efficiency

Average running time is computed on the first 1,000 images of the MSRA10K dataset. We choose the five methods with the closest performances to the proposed approach in the test, and the results are shown in Table 3.2. The proposed algorithm is significantly faster than CB, LR and PCA; and although being slower than MR and GS, our method still outperforms them both considering the overall evaluation performances.

Table 3.2 Running time test results of selected methods (seconds per image)

Method	Ours	CB	GS	MR	PCA	LR
Time(s)	1.12	1.71	0.324	0.869	3.15	13.8

3.5.8 Limitation

One limitation of our proposed RR method is that the erroneous boundary removal step is still based on major voting, i.e. when more than two of the four boundaries are actually covered by the foreground object, the foreground will become major, while the background will become minor. Such phenomenon will result in the less significant background boundary be treated as “foreground” as be removed, which will lead to completely reversed saliency maps. This issue, however, will only emerge occasionally since the images with most boundaries covered by the foreground object are less common. On most cases, our method still prevails over the comparison state-of-the-art methods in the overall performance.

3.6 Summary

In this chapter, we propose a novel bottom-up saliency detection method with erroneous boundary removal and regularized random walks ranking. There are two major

innovation aspects: firstly, the erroneous boundary removal process effectively eliminates the image boundary with boundary-adjacent foreground superpixels, and thus neutralizes their negative influences in the saliency estimations; secondly, the proposed regularized random walks ranking provides prior saliency estimation to all pixels in the input image, which leads to pixel-wisely detailed and superpixel-independent saliency map outputs. Our approach is fully-automatic without any user supervision requirement. Results of experiments on two public datasets show that the proposed method significantly outperforms twelve state-of-the-art saliency detection algorithms in terms of both accuracy and robustness, as well as maintaining high efficiency compared with other methods.

Chapter 4 Improved Low-Level Feature Based Saliency Detection

In the previous chapter, we have introduced the RR method, which is a novel saliency detection method based on conventional low-level features. RR has exhibited higher performance against state-of-the-art methods; nevertheless, there are still limitations that restrict RR from its full potential.

In this chapter, to further improve the performance, we present the reversion correction and regularized random walks ranking (RCRR) method, which is based on the RR method but has significant technical innovations. Section 4.1 summarizes the challenge we are going to address; section 4.2 lists our contributions and the major steps in our proposed RCRR method; section 4.3 reviews the *k*-means clustering algorithm, which is an additional related work besides the manifold ranking and random walks in section 3.3; section 4.4 gives step-by-step methodology of RCRR; section 4.5 includes experimental results and discussion; and finally, section 4.6 concludes this chapter.

4.1 Problem Formulation

Studies in [1], [54] and [52] show that boundary-based bottom-up saliency detection algorithms are becoming popular according to related state-of-the-art researches. These algorithms are generally facilitated by superpixel segmentation, and their results outperform most of the other state-of-the-art saliency detection algorithms. Nevertheless, there still exist drawbacks that hinder these algorithms, with two major issues as below:

(1) It may be implausible to directly apply four image boundaries as the background queries for the background saliency detection. More specifically, one or more of the boundaries may contain part of the foreground object, and undesired error may occur if they are still considered as the background. Examples are shown in Figure 4.1, where the salient objects take considerable parts of the image boundaries, leading to the failure of the MR method [52]. We also note that due to the negative influences of the boundary-adjacent foreground objects, the saliency maps in Figure 4.1(b) look similar to the "reversed" version of the ground truth in Figure 4.1(d), i.e., most of the background regions are classified as foreground, and most of the foreground regions are classified as background.

(2) The superpixel segmentation [166] facilitates the pre-processing of boundary-based (and many other graph-based) saliency detection algorithms. However, inaccuracy in the superpixel segmentation itself may directly lead to the failure of the entire algorithm. Moreover, the operation of assigning the same saliency value to all the pixels within a fix-sized patch unavoidably ignores some detailed information from the original image, making the saliency map as if being covered by mosaics, and hence lowering the overall visual quality. It is thus desirable to combine both superpixel-wise and pixel-wise image data in the saliency detection, in which the pixel-wise process can provide better smoothness and hence improve the overall quality and accuracy of the output saliency map.

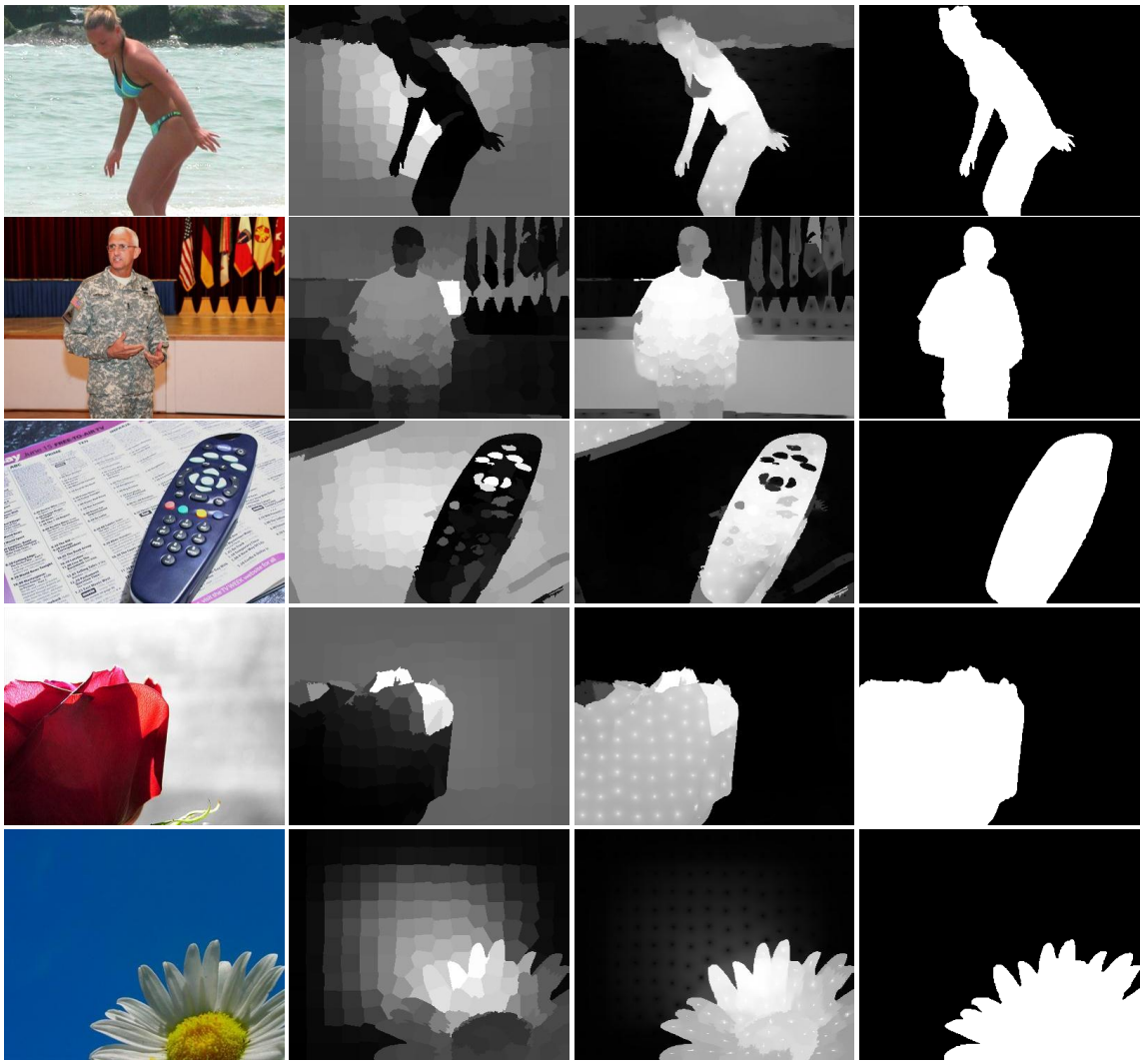

Figure 4.1 Examples showing the problem of using boundaries as background queries when the salient objects are boundary-adjacent. (a) Input images; (b) results of a boundary-based method [52]; (c) results of our proposed RCRR method; (d) ground truth. Our method can effectively prevent the "saliency reversion" problem.

4.2 Contributions

In this chapter, in order to overcome the two issues above, we propose the reversion correction and regularized random walks ranking (RCRR) for saliency detection, a novel graph-based bottom-up saliency detection method. Our key contributions are summarized below:

(1) We present the reversion correction (RC) process, which locates and eliminates the boundary-adjacent foreground superpixels, preventing the saliency reversions from emerging, such as the cases in Figure 4.1(b). This mechanism provides increased robustness, as shown in Figure 4.1(c).

(2) We build the regularized random walks ranking (RRWR) model, which takes both prior saliency estimations and pixel-wise image data into account. RRWR is independent of superpixel segmentation, and is able to generate pixel-wise saliency maps that reflect full-details of the input images.

(3) We explore the extensibility of RC as an optimization algorithm on existing boundary based saliency detection methods, which has the potential of significant performance boosting.

(4) We also propose the boundary-adjacent object saliency (BAOS) dataset, which contains 200 images that have large proportions of the salient objects on the image boundaries. This dataset provides an objective evaluation for saliency detection methods' performances on boundary-adjacent salient objects.

This work is an extension to our previous study [1] with marked improvements, especially the technical contributions above. In addition, we have conducted a more detailed and comprehensive evaluation with 14 state-of-the-art methods, including our previous work [1], on five datasets. The results imply the superiority of our proposed RCRR method in terms of both accuracy and robustness.

4.3 Related Works

Our proposed RCRR method is based on manifold ranking, random walks, and *k*-means clustering. Section 3.3 has already introduced the basic principles of manifold ranking

and random walks. In this section, we will briefly introduce the *k*-means clustering algorithm as supplementary knowledge.

4.3.1 *K*-Means Clustering

The k-means clustering partitions the elements in χ into K clusters $S = \{S_1, S_2, ..., S_K\}$, on the condition that the within-cluster sum of squared error is minimized:

$$S = \arg\min_{S} \left(\sum_{k=1}^{K} \sum_{x \in S_k} \| x - m_k \|^2 \right), \tag{4.1}$$

where m_k is the mean of observations in S_k.

In the proposed algorithm, given its efficiency, robustness and accuracy, the *k*-means clustering is used to group the initial saliency estimation result into foreground / background clusters. The boundary-adjacent foreground superpixels are then recognized and removed. Detailed steps are presented in section 4.4.1.

4.4 Saliency Detection with Reversion Correction and Regularized Random Walks Ranking (RCRR)

Our saliency detection algorithm (RCRR) consists of two major steps. The first step comprises the saliency reversion correction (RC) process on an initial saliency estimation, which eliminates the boundary-adjacent foreground regions from the image boundaries; the second step extracts seed references from the first step, and calculates the final pixel-wise saliency map with regularized random walks ranking (RRWR).

4.4.1 Saliency Reversion Correction

As stated in section 4.1, it is possible that the foreground object is on one or more boundaries of the input image. Using such problematic boundaries as queries in the saliency estimation may lead to undesirable results. Typical examples are illustrated in Figure 4.1(b), where, due to the negative influences of the boundary-adjacent foreground superpixels, the corresponding saliency maps are nearly "reversed" in comparison with the ground truth in Figure 4.1(d). To address this issue, it is tempting to directly conduct classification among all the boundary superpixels; however, such classification with the boundary information alone may be too subjective without the global context. We thus propose the reversion correction (RC) process, which functions as a posterior classification method based on initial saliency estimation. The boundary-adjacent foreground regions will then be detected and removed, improving the overall robustness of our algorithm.

For an input image, we first obtain an initial saliency estimation, which can be generated by any boundary-based saliency detection method (e.g. [52], [54], [80]). The graph-based manifold ranking [52] is used in our method due to its relatively high performance and efficiency. With (3.3), the initial saliency estimation is acquired as:

$$S_{init}(i) = f^*(i), i = 1,...,n, \tag{4.2}$$

where n is the number of superpixels in the image. S_{init} is then partitioned into background/foreground superpixels by *k*-means clustering with Lloyd's algorithm [170]:

(1) Two uniformly-distributed mean values of S_{init} are generated: $m_k = \frac{k}{3}\max(S_{init}), k = 1, 2$;

(2) Associate each element of S_{init} to one of the two clusters with the closest mean value m_k;

(3) Each m_k is then replaced by the mean saliency value of all the elements just assigned to the corresponding cluster;

(4) Repeat steps (2) and (3) until a convergence of the two clusters or a desired number of iteration is reached. The labeling map L_{kmeans} is obtained afterwards.

In L_{kmeans}, background superpixels are labeled with 1, while foreground superpixels are labeled with 2. The next step is to recognize if S_{init} is "reversed". Empirically there are less (or no) foreground superpixels on the boundaries of most "normal" saliency maps; if the majority (or all) of the boundaries of a saliency map are covered with foreground superpixels, we may confidently assume it as "reversed". Therefore, we calculate the average label L_b of all the boundary-adjacent superpixels in L_{kmeans}; if L_b is greater than a pre-defined threshold $t_{reverse}$, we will treat S_{init} as reversed.

The following step is based on the judgment of S_{init}:

(1) If S_{init} is determined as reversed, we will find and remove all of the boundary-adjacent superpixels under the guidance of L_{kmeans}, i.e. remove all of the background (marked as 1) boundary superpixels in L_{kmeans}, because due to the saliency reversion, they are actually the foreground superpixels. And then, the initial saliency estimation step is re-performed with the newly formed boundary queries.

(2) If S_{init} is determined as not reversed, nothing will be conducted.

The workflow of RC is summarized in Table 4.1.

Table 4.1 Algorithm description of the RC process

Step	Content
Input	Initial saliency estimation S_{init}, threshold $t_{reverse}$.
1	Calculate L_{kmeans}. The background and foreground superpixels are labeled with 1 and 2, respectively.
2	Calculate the average boundary label L_b.
3	If $L_b \geq t_{reverse}$, locate and remove the boundary superpixels of S_{init} with label 2 on L_{kmeans}; Repeat the initial saliency estimation with refined boundary to obtain the updated result S_{RC}.
4	If $L_b < t_{reverse}$, directly output $S_{RC} = S_{init}$.
Output	The saliency estimation after RC S_{RC}.

The major advantage of RC is that it directly counters the source of the saliency reversion, i.e. the boundary-adjacent foreground superpixels. By locating and eliminating the boundary-adjacent foreground superpixels, their negative influences can be neutralized, which reverses the "reversed" saliency map back to normal, as shown in Figure 4.2(c). In addition, nothing will be done if the initial saliency estimation is detected as normal, which ensures that no further error will be introduced by RC to the good results.

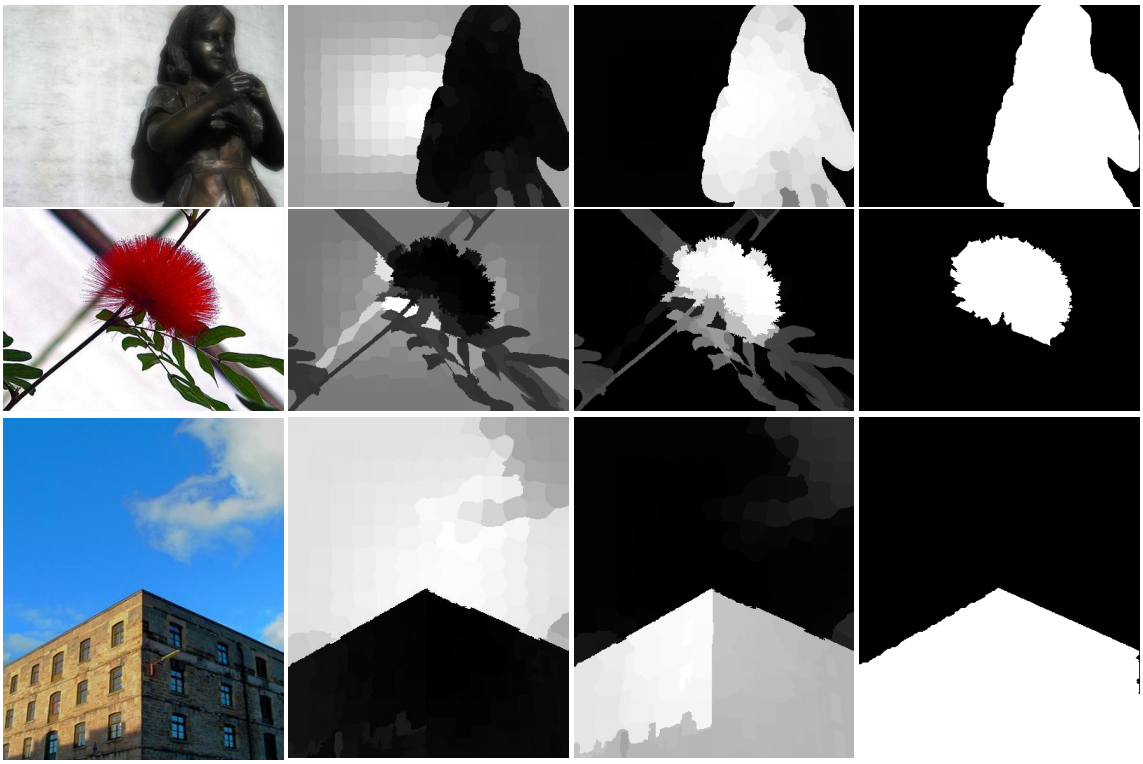

Figure 4.2 Examples of RC. (a) Input images; (b) saliency estimations without RC; (c) saliency estimations with RC; (d) ground truth. The RC step can effectively counteract the saliency reversion problem due to the boundary-adjacent objects.

4.4.2 Regularized Random Walks Ranking

As introduced in section 1.3, many state-of-the-art saliency detection algorithms (e.g. [52-54]) heavily depend on the pre-processed superpixel segmentation, which may generate undesirable results if the superpixel segmentation itself is imprecise. Besides that, assigning the same saliency value to all pixels within a superpixel sacrifices the detail information from the original image. To overcome these drawbacks, we develop the regularized random walks ranking (RRWR) model, which is independent of the superpixel segmentation, and can reveal accurate pixel-wise saliency of the input image.

RRWR is initially proposed in our previous study [1]. It is based on (3.8), but we suggest a new fitting constraint, which restricts the Dirichlet integral to be as close to the prior saliency distribution as possible:

$$Dir\left[p^l\right]=\frac{1}{2}\left(p^l\right)^T L\left(p^l\right)+\frac{\eta}{2}(p^l-Y)^T(p^l-Y) \tag{4.3}$$

where the second term $\frac{\eta}{2}(p^l-Y)^T(p^l-Y)$ is the newly added fitting constraint, η is a controlling parameter, similar to the μ used in (3.2), and Y is a pixel-wise indication vector inheriting the values of S_{RC} from section 4.4.1. Note that RRWR is computed pixel-wisely, hence both p^l and Y are $N\times 1$ vectors, and L is an $N\times N$ matrix, where N is the total pixel number in the image. We define two thresholds t_{high} and t_{low} as:

$$\begin{aligned} t_{high} &= \frac{\text{mean}(S_{RC})+\max(S_{RC})}{2} \\ t_{low} &= \text{mean}(S_{RC}) \end{aligned} \tag{4.4}$$

which are used to select pixels with $Y_u > t_{high}$ as foreground seeds, and $Y_u < t_{low}$ as background seeds. The seeds are then combined into $p_M^l, l=1,2,...$ in section 3.3.2, where $l=1$ corresponds to the background label, and $l=2$ corresponds to the foreground label. The matrix decomposition of (4.3) is conducted as below:

$$\begin{aligned} Dir\left[p^l\right] &= \frac{1}{2}\left[\left(p_M^l\right)^T \left(p_U^l\right)^T\right]\begin{bmatrix} L_M & B \\ B^T & L_U \end{bmatrix}\begin{bmatrix} p_M^l \\ p_U^l \end{bmatrix} \\ &+\frac{\eta}{2}\left(\begin{bmatrix} p_M^l \\ p_U^l \end{bmatrix}-\begin{bmatrix} Y_M^l \\ Y_U^l \end{bmatrix}\right)^T\left(\begin{bmatrix} p_M^l \\ p_U^l \end{bmatrix}-\begin{bmatrix} Y_M^l \\ Y_U^l \end{bmatrix}\right) \end{aligned} \tag{4.5}$$

Similar to (3.9), after setting the differentiation of (4.5) with respect to p_U^l as zero, the optimal solution is obtained as:

$$p_U^l = \left(L_U + \eta I\right)^{-1}\left(-B^T p_M^l + \eta Y_U^l\right). \tag{4.6}$$

Then p_U^l and p_M^l are united as p^l. We set $l = 2$ to select the foreground possibility p^2, and reshape it to a matrix S_{final} with same size of the input image as the final foreground saliency output.

Since the seeds are automatically generated from the result of the RC, unlike classical random walks [117], no user interaction is required in RRWR. The fitting constraint in (4.3) provides a prior saliency estimation to all pixels instead of the seed pixels alone, which offers a better guidance in calculating the final saliency map. The effect of the fitting constraint is shown in Figure 4.3, where RRWR (Figure 4.3(d)) not only improves the saliency map from the initial saliency estimation (Figure 4.3(b)), but also remarkably outperforms classical random walks (Figure 4.3(c)), which uses the first term of (4.3)) alone.

The complete workflow of our proposed RCRR method is listed in Table 4.2.

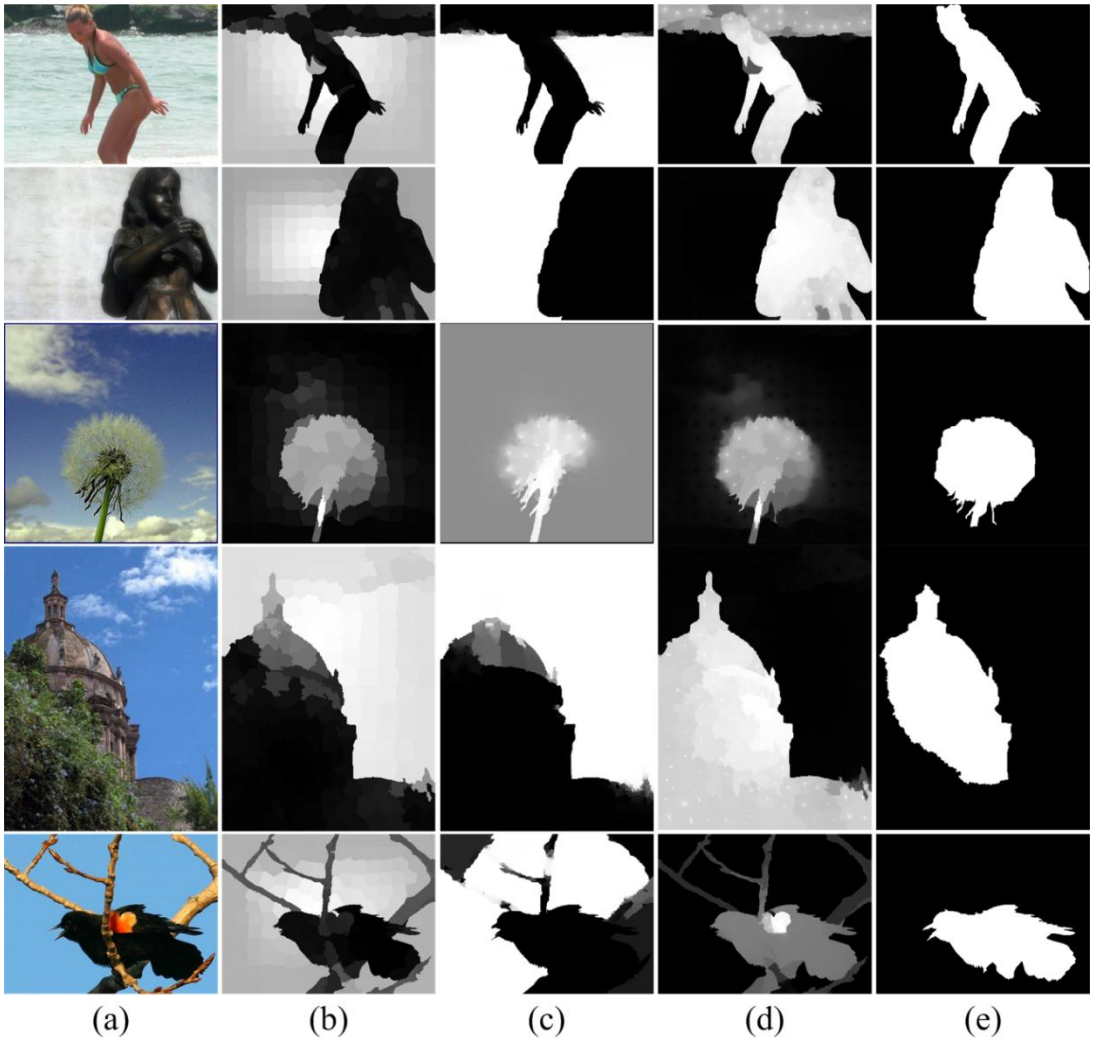

Figure 4.3 Examples of RRWR. (a) Input images; (b) initial saliency estimations; (c) saliency outputs with classical random walks; (d) saliency outputs with RRWR; (e) ground truth. RRWR is able to further refine the initial saliency estimations. Column (c) and (d) shows that RWRR remarkably outperform the classical random walks.

Table 4.2 Algorithm description of our proposed RCRR method

Step	Content
Input	An image and related parameters.
1	Establish superpixel graph; calculate W and D.
2	Conduct initial saliency estimation and obtain S_{init}.
3	Conduct RC in Table 4.1 and obtain S_{RC}.
4	Compute the pixel-wise saliency p^l with (4.6).
5	Set $l=2$ and reshape p^2 into S_{final}.
Output	A saliency map with the same size of the input image.

4.5 Experimental Results

In this section, we present the experimental results of our proposed RCRR method. We first introduce the datasets, evaluation metrics and algorithm parameters we used, and then evaluate the two design options in our method (i.e. RC and RWRR). After that, we present the comparison experiment against fourteen state-of-the-art saliency detection methods, with both quantitative and qualitative analyses. We also explore the extensibility of our method as a saliency optimization algorithm, which is conducted on any existing saliency detection method to further refine its performance. Finally, we present the efficiency and limitation of our method.

4.5.1 Datasets

Our experiments are conducted on five datasets, including four publicly available datasets and one newly designed dataset. The four public datasets (based on a recent saliency detection benchmark [20]) are:

(1) MSRA10K [64], which contains 10,000 randomly-chosen images from the MSRA dataset [8];

(2) ECSSD [69], which contains 1,000 complex natural images with diversified patterns;

(3) SED [171], which contains 100 images with one salient object and 100 images with two salient objects (200 images in total); and

(4) PASCAL-S [172], which ascends from the PASCAL VOC [173] segmentation challenge and contains 850 images with complex background.

We also propose and use the new boundary-adjacent object saliency (BAOS) dataset, which is specifically designed to evaluate the images where large portions (at least 30%)

of their boundaries are covered by the foreground object(s). It contains 200 images (selected from MSRA10K, ECSSD, and Microsoft Grabcut [115]).

All of the datasets come with human-labeled pixel-wise ground truth.

4.5.2 Evaluation Metrics

We follow the existing metrics in [20], and use precision-recall curve, F-measure, and mean absolute error (MAE) score as our evaluation metrics. The terms of precision, recall and F-measure are defined in [168] as:

$$precision = \frac{\sum_{i=1}^{N} G(i) \cdot I\left(S_{final}(i) \geq \text{th}\right)}{\sum_{i=1}^{N} I\left(S_{final}(i) \geq \text{th}\right)} \tag{4.7}$$

$$recall = \frac{\sum_{i=1}^{N} G(i) \cdot I\left(S_{final}(i) \geq \text{th}\right)}{\sum_{i=1}^{N} G(i)} \tag{4.8}$$

$$F_{\beta} = \frac{(1+\beta^2)\, precision \cdot recall}{\beta^2\, precision + recall} \tag{4.9}$$

where G is the ground truth; $I(\cdot)$ is the indicator function that equals to 1 if the condition inside is satisfied, and 0 otherwise; S_{final} is the output saliency map corresponding to Algorithm 2; *th* is the threshold used to binarize S_{final}; and N is the number of pixels in the image. Precision and recall are usually displayed together as precision-recall curves, which are constructed by binarizing the saliency map with thresholds changing from 0 to 255. The F-measure is adopted as a weighted average between precision and recall. As suggested in [174], the average F-measure of a precision-recall curve is computed as its maximal single-point F-measure. We set

$\beta^2 = 0.3$ to grant more importance to the precision, which is consistent to [20]. When used to evaluate a saliency map, the higher the evaluation metric (precision, recall or f-measure), the better the estimation.

On the other hand, MAE is defined as the mean of the difference between the saliency map and the ground truth:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|S_{final}(i) - G(i)\right| \tag{4.10}$$

Note that different to the previous evaluation metrics, it is smaller MAE that means better estimation.

In addition, to evaluate the statistical significance level of RCRR against a comparison method A, we conduct Student's *t*-test between the two methods. We equally divide the images of a particular dataset into 10 subgroups and compute the evaluation metric (F-measure or MAE) in each group. This enables us to obtain the sample mean and sample standard deviation of RCRR and A, namely $\overline{X}_{RCRR}$, $\overline{X}_{A}$, $s_{X_{RCRR}}$ and s_{X_A}. The *t*-statistic is then computed as:

$$t = \frac{\overline{X}_{RCRR} - \overline{X}_{A}}{s_{X_{RCRR}X_A} \cdot \sqrt{\frac{2}{10}}} \tag{4.11}$$

where

$$s_{X_{RCRR}X_A} = \sqrt{\frac{s_{X_{RCRR}}^2 + s_{X_A}^2}{2}} \tag{4.12}$$

We then find the one-sided *p*-value corresponding to *t* with 10-1=9 as the degree of freedom, since our alternative hypothesis is that the metric from RCRR is significantly

larger (F-measure) or lower (MAE) than that of A, but not both. The *p*-value is given together with its corresponding evaluation metric in our experiments.

4.5.3 Parameters

To objectively compare our algorithm with other algorithms, we use the same parameter settings as in [52], where the superpixel number is set to $n = 200$, and the two controlling parameters in (3.4) and (3.3) are set to $\sigma^2 = 0.1$ and $\mu = 0.01$, respectively. The only new parameter in RC is the average boundary label threshold $t_{reverse}$, which is one of the inputs of Table 4.1. We empirically set $t_{reverse} = 1.5$, which results in the peak performance in Figure 4.4. And the only new parameter in RRWR is the controlling parameter η. We empirically set $\eta = 0.01$, which results in the peak performance in Figure 4.5.

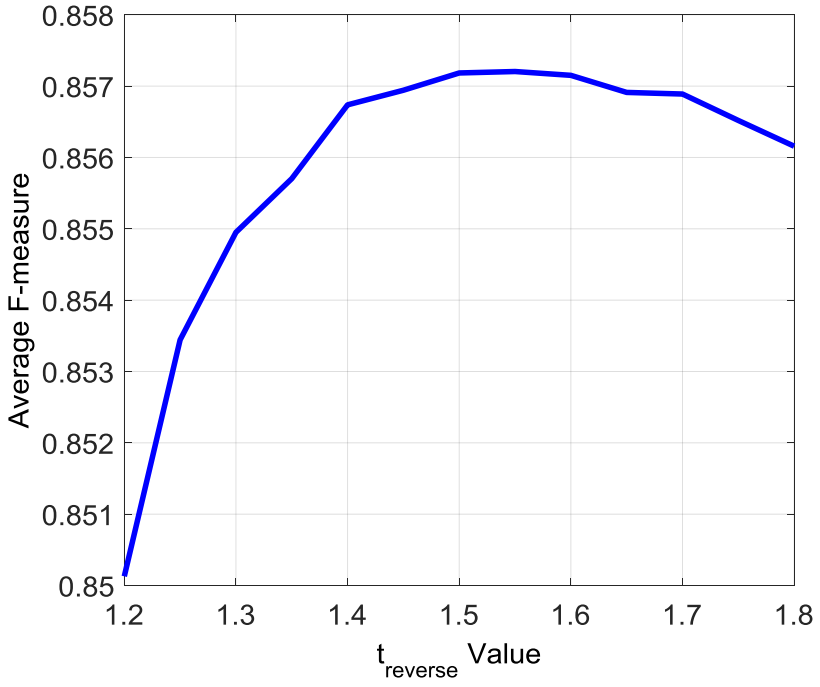

Figure 4.4 Average F-measures with different $t_{reverse}$ used in RC on the MSRA10K dataset. The value $t_{reverse} = 1.5$, which corresponds to the optimal F-measure, is adopted in our following experiments.

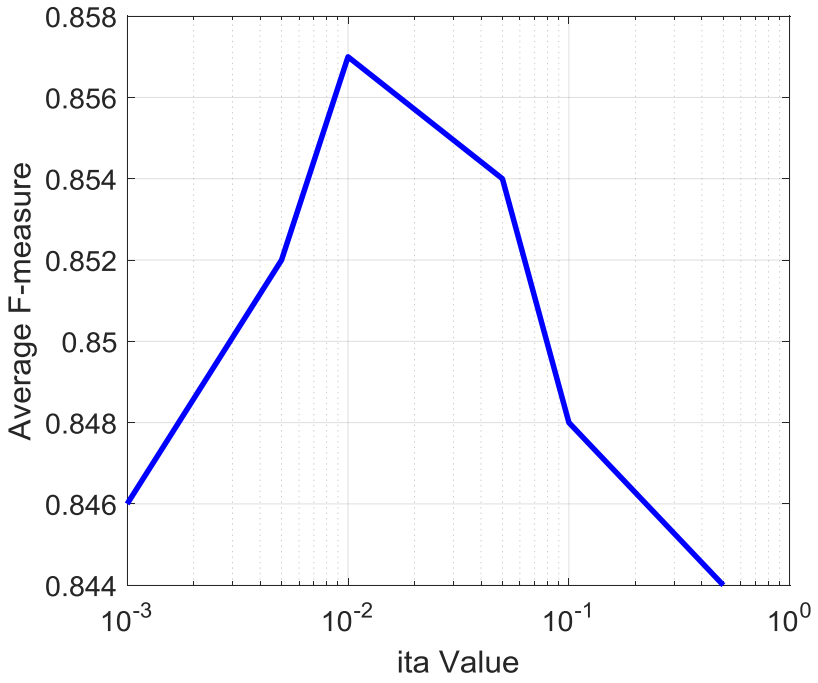

Figure 4.5 Average F-measures with different η used in RRWR on the MSRA10K dataset. The value $\eta = 0.01$, which corresponds to the optimal F-measure, is adopted in our following experiments.

4.5.4 Implementation

Our method is implemented in MATLAB on a 64-bit PC with Intel 6-Core i7-5820K CPU @ 3.3GHz and 64GB RAM. The source code of the RCRR method, together with the BAOS dataset, are both available at online: https://github.com/yuanyc06/rcrr/.

4.5.5 Evaluation of Design Options

We first examine the contributions of our algorithm, namely RC and RRWR. The red and blue curves in Figure 4.6 show the improvements in the precision-recall curves with the use of RC when compared to the saliency output without RC. Similarly, Figure 4.7 exhibits that the F-measure of our method (0.857) is higher than that without using RC (0.850). After that, we generate the saliency maps without the use of RRWR. As shown by the red and brown curves in Figure 4.6, the complete algorithm exhibits superiority

over the algorithm without RRWR. We also notice in Figure 4.7 that our proposed method achieves a higher F-measure than that without RRWR, in which the values are 0.857 in comparison with 0.848, respectively.

Based on the analyses above, both RC and RRWR have contributions in improving the overall performance.

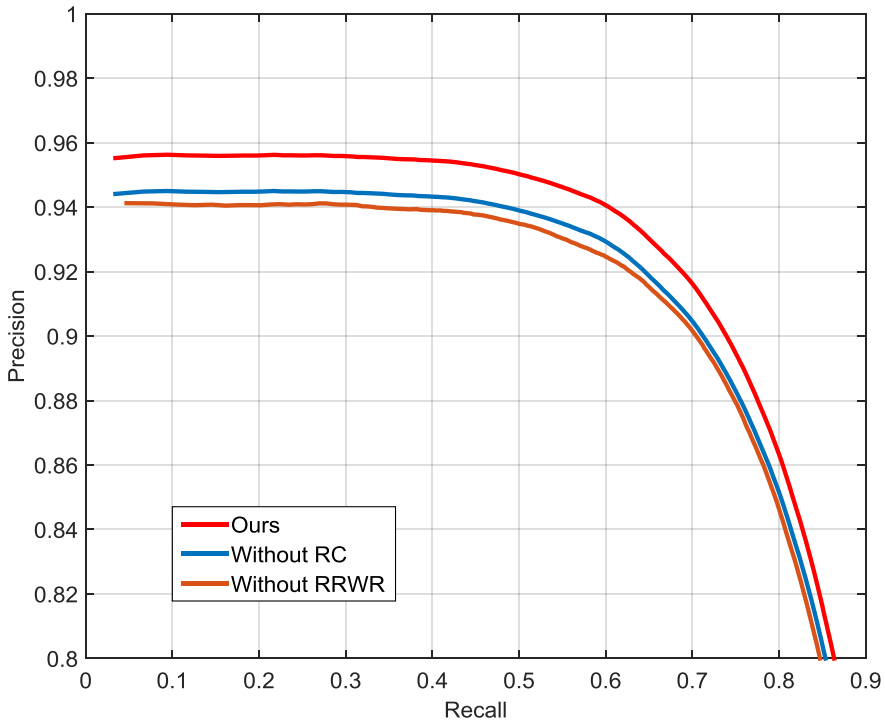

Figure 4.6 The precision-recall curves of our method, our method without using RC, and our method without using RRWR.

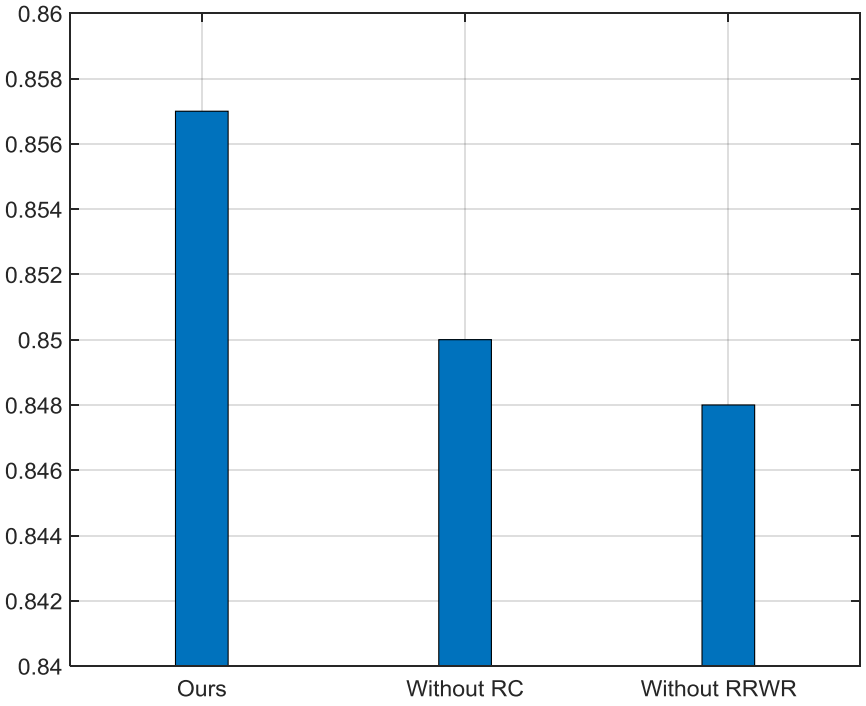

Figure 4.7 The average F-measures of our method, our method without using RC, and our method without using RRWR.

4.5.6 Comparison with State-of-the-Art

We evaluate the proposed RCRR method on the five datasets introduced in section 4.5.1, in comparison with fourteen state-of-the-art saliency detection methods, namely CA [78], CB [75], DSR [80], FES [76], FT [66], HS[69], IT[5], LR[77], MC [54], MR [52], RR [1], SF [70], SR [67], and wCtr* [53]. All of the algorithms above are evaluated by the corresponding authors' online available software codes. Our evaluation is conducted both quantitatively and qualitatively. Note that all of the methods above (including our RCRR method) are non-training-based. Other methods such as DRFI [57] are excluded as they require additional training data, the choice of which will significantly affect their performances.

Quantitative Evaluation:

The complete quantitative evaluation results are summarized in Table 4.3, and detailed analyses of individual datasets are presented in Figure 4.8 to Figure 4.22.

We first conduct our quantitative evaluation on the MSRA10K dataset, which is large enough to cover most types of natural images. The results are shown in Figure 4.8 to Figure 4.10. It is obvious that our method excels all of the other methods among the precision-recall curves in Figure 4.8, where its highest precision value reaches up to 0.96. Our method also achieves the best F-measure 0.857 in Figure 4.9, and the second best MAE score 0.117 in Figure 4.10. In addition, the *p*-values on Figure 4.9 and Figure 4.10 also indicate that the advantages of RCRR against the comparison methods in both F-measure and MAE are statistically significant.

Table 4.3 F-measure and MAE evaluation results.

	F-measure					**MAE**				
	MSRA10K	**ECSSD**	**SED**	**PASCAL-S**	**BAOS**	**MSRA10K**	**ECSSD**	**SED**	**PASCAL-S**	**BAOS**
CA	0.621	0.513	0.603	0.496	0.605	0.237	0.343	0.246	0.301	0.392
CB	0.764	0.672	0.693	0.625	0.665	0.209	0.289	0.254	0.286	0.362
DSR	0.834	0.699	0.806	0.651	0.693	0.121	0.226	0.151	0.215	0.335
FES	0.717	0.618	0.672	0.624	0.613	0.185	0.265	0.207	0.223	0.389
FT	0.583	0.426	0.605	0.406	0.545	0.242	0.329	0.247	0.316	0.412
HS	0.845	0.698	0.806	0.645	0.729	0.149	0.269	0.179	0.264	**0.303**
IT	0.480	0.415	0.507	0.421	0.483	0.217	0.285	0.233	0.246	0.413
LR	0.773	0.631	0.720	0.580	0.691	0.225	0.313	0.247	0.288	0.365
MC	0.847	0.703	0.810	0.658	0.684	0.145	0.251	0.172	0.232	0.346
MR	0.846	0.708	0.802	0.612	0.711	0.126	0.236	0.154	0.259	0.330
RR	0.850	**0.710**	0.806	0.639	**0.737**	0.121	0.229	0.151	0.232	0.306
SF	0.749	0.549	0.719	0.496	0.670	0.171	0.268	0.202	0.241	0.382
SR	0.528	0.450	0.541	0.454	0.570	0.249	0.345	0.253	0.294	0.407
wCtr*	**0.853**	0.687	**0.815**	**0.659**	0.724	**0.112**	**0.225**	**0.147**	**0.208**	0.330
Ours	**0.857**	**0.714**	**0.811**	**0.663**	**0.742**	**0.117**	**0.223**	**0.150**	**0.212**	**0.296**

*The best and second best results are marked in **red** and **blue**, respectively.*

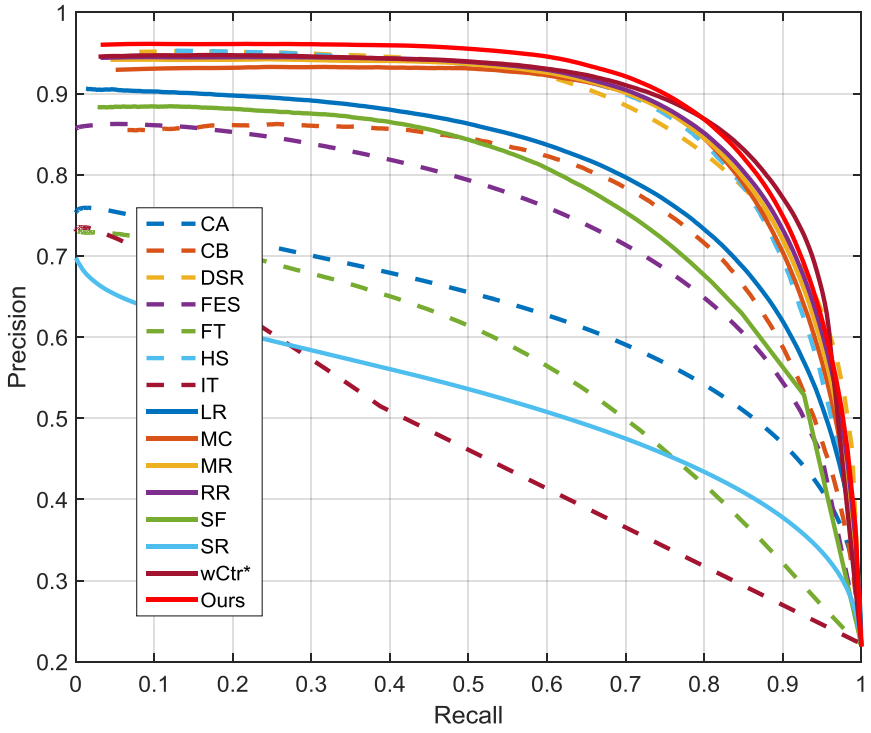

Figure 4.8 Precision-recall curves on the MSRA10K dataset.

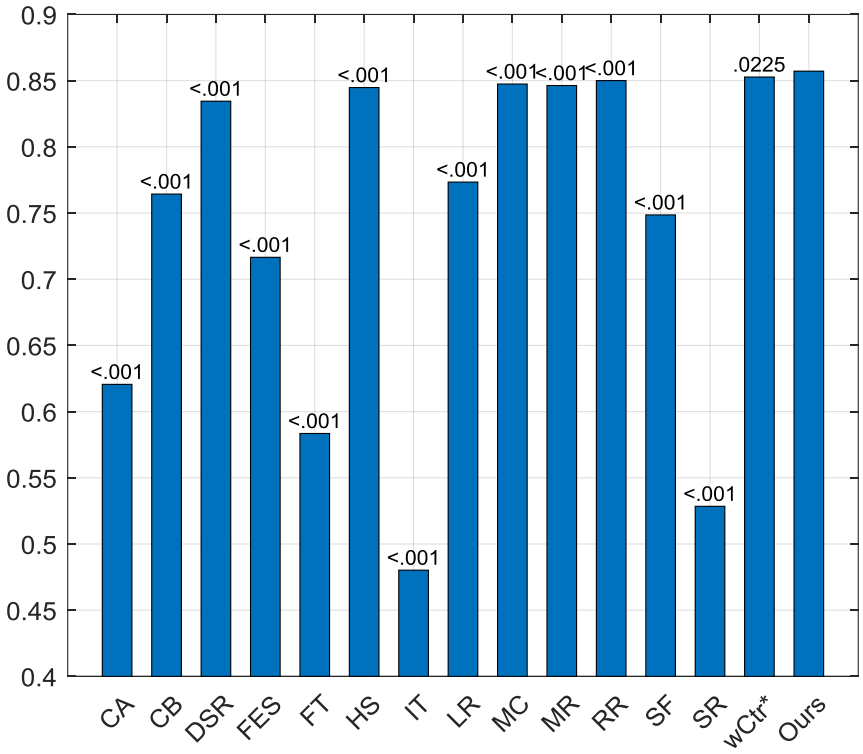

Figure 4.9 F-measures on the MSRA10K dataset.

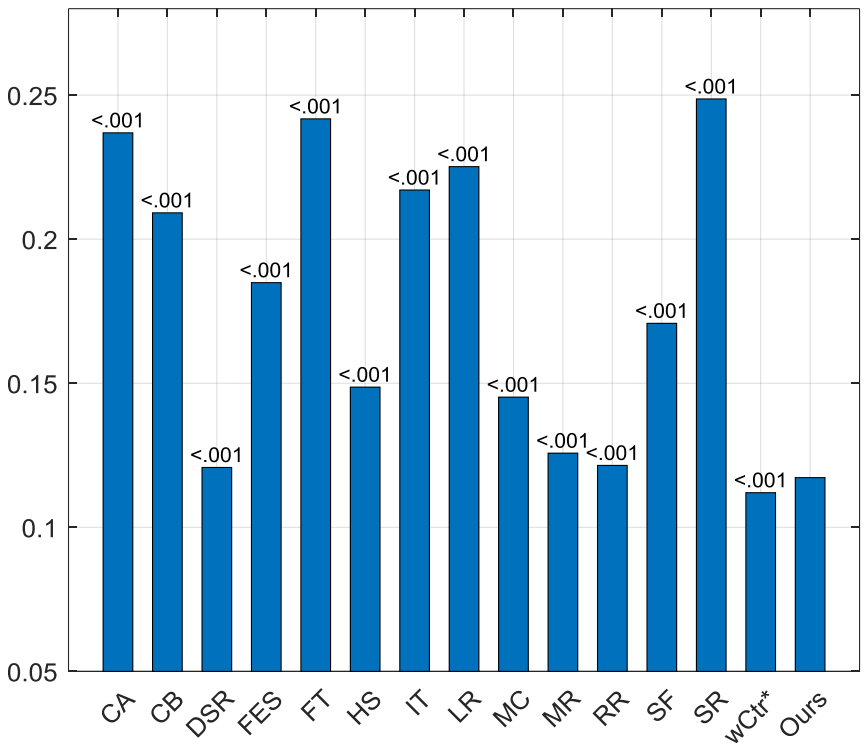

Figure 4.10 MAE scores on the MSRA10K dataset.

And then we proceed to the ECSSD dataset, which contains 1,000 images with complicated backgrounds. Again, our method outperforms all of the other methods among the precision-recall curves in Figure 4.11. This observation is further validated in Figure 4.12 and Figure 4.13, where our method achieves the highest F-measure 0.711 and the lowest MAE score 0.224 simultaneously, with statistically significant advantages.

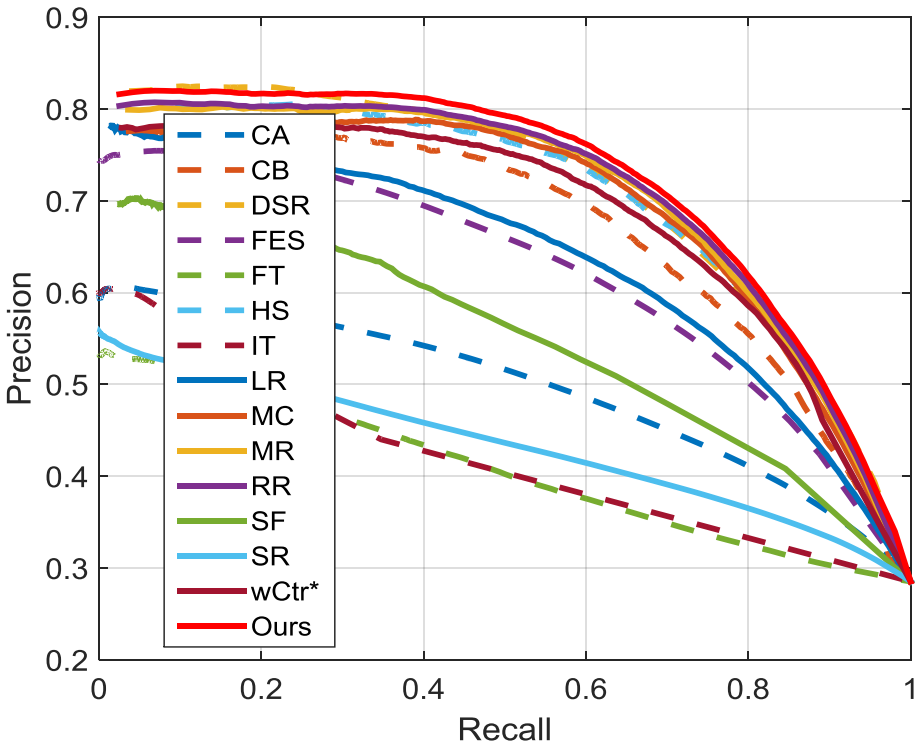

Figure 4.11 Precision-recall curves on the ECSSD dataset.

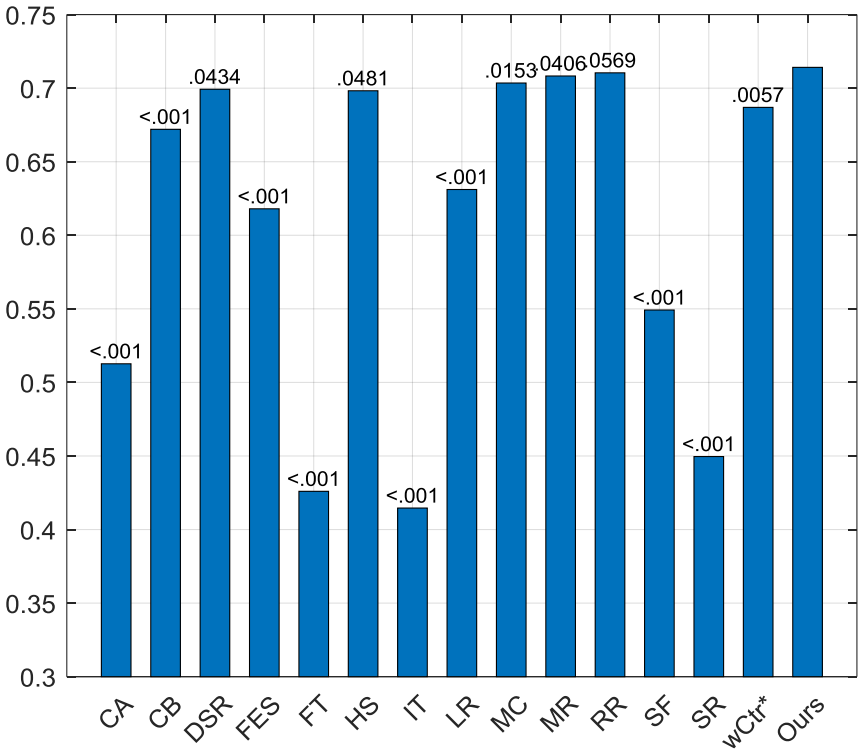

Figure 4.12 F-measures on the ECSSD dataset.

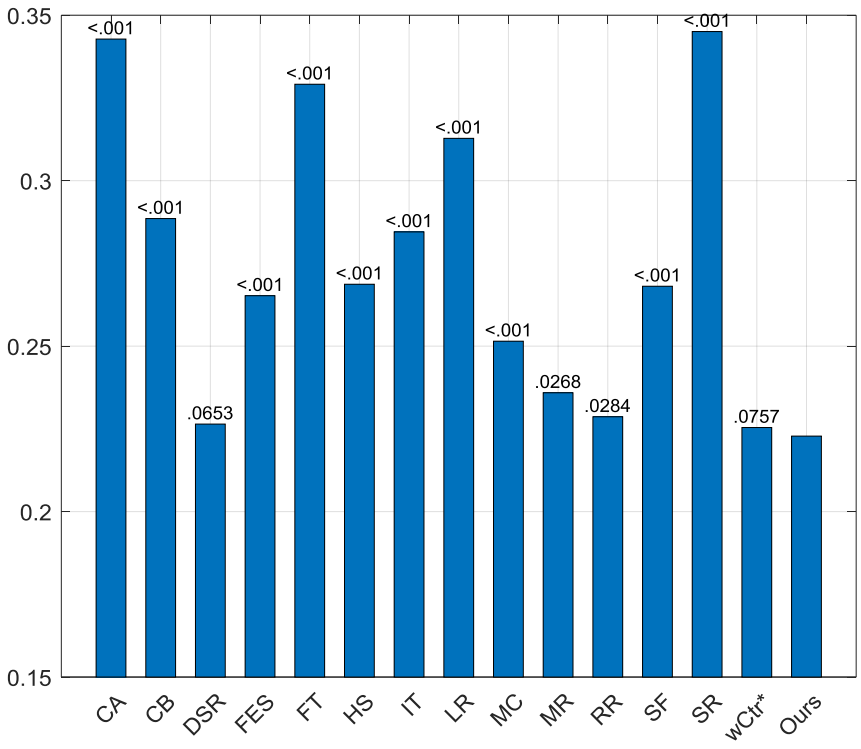

Figure 4.13 MAE scores on the ECSSD dataset.

Our method behaves similarly on the SED dataset and the PASCAL-S dataset (Figure 4.14 to Figure 4.19), where it outperforms most of the comparison methods among the precision-recall curves, and only marginally worse to wCtr* at some points. Our method, DSR, HS and wCtr* have entangled curves in Figure 4.14 and Figure 4.17, and have close scores in F-measure and MAE. Nevertheless, our method still achieves the best F-measure (0.663) on PASCAL-S, the second best F-measure (0.811) on SED, and the second best MAE scores (0.150 and 0.212) on SED and PASCAL-S, respectively. The statistical *p*-values of our method on SED and PASCAL-S are not as significant as those on MSRA10K and ECSSD, which match the mixed performances we observed above; but our *p*-values still maintain being under 0.1.

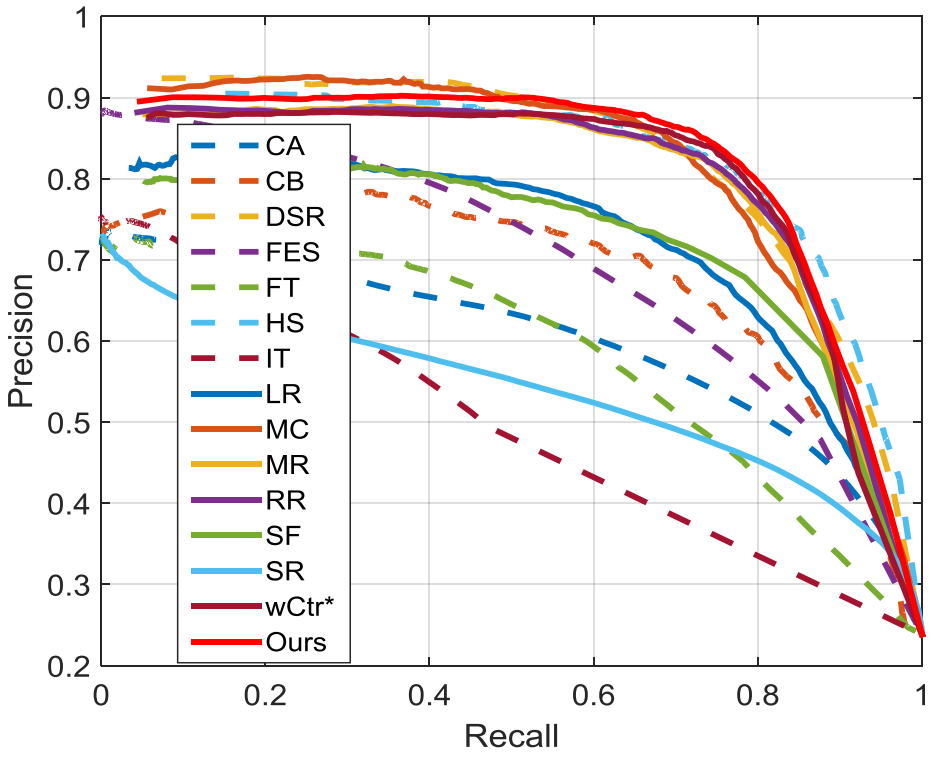

Figure 4.14 Precision-recall curves on the SED dataset.

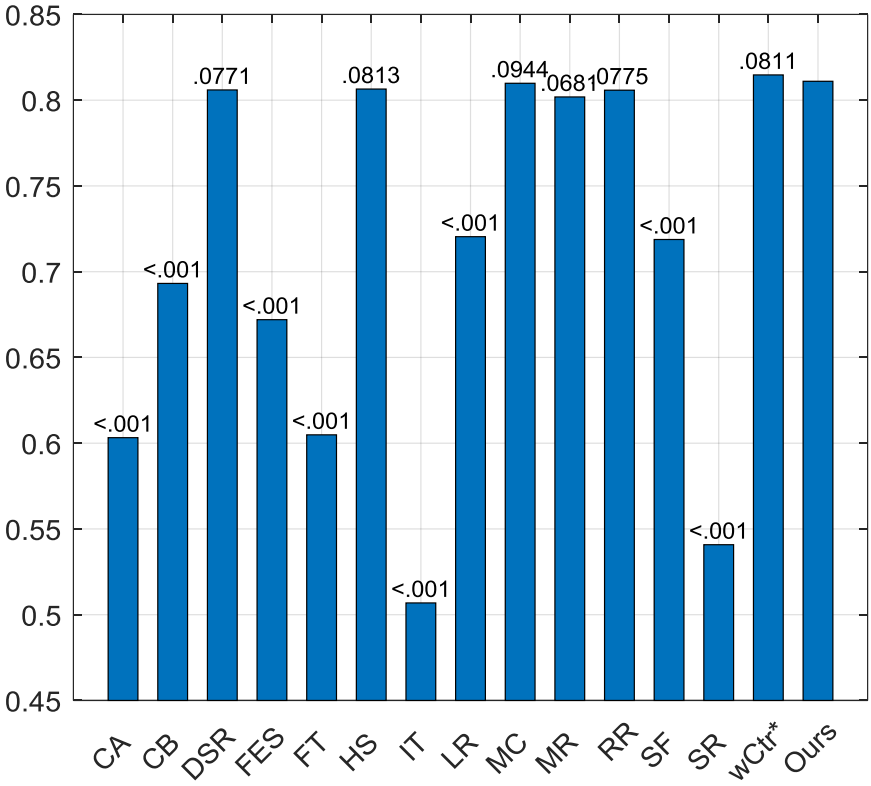

Figure 4.15 F-measures on the SED dataset.

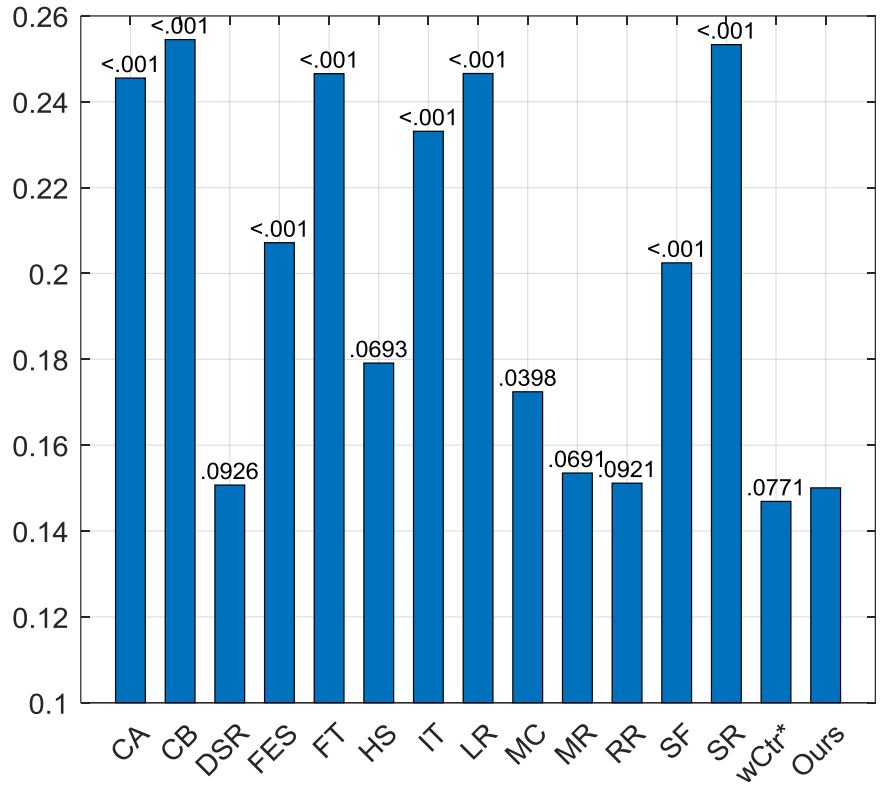

Figure 4.16 MAE scores on the SED dataset.

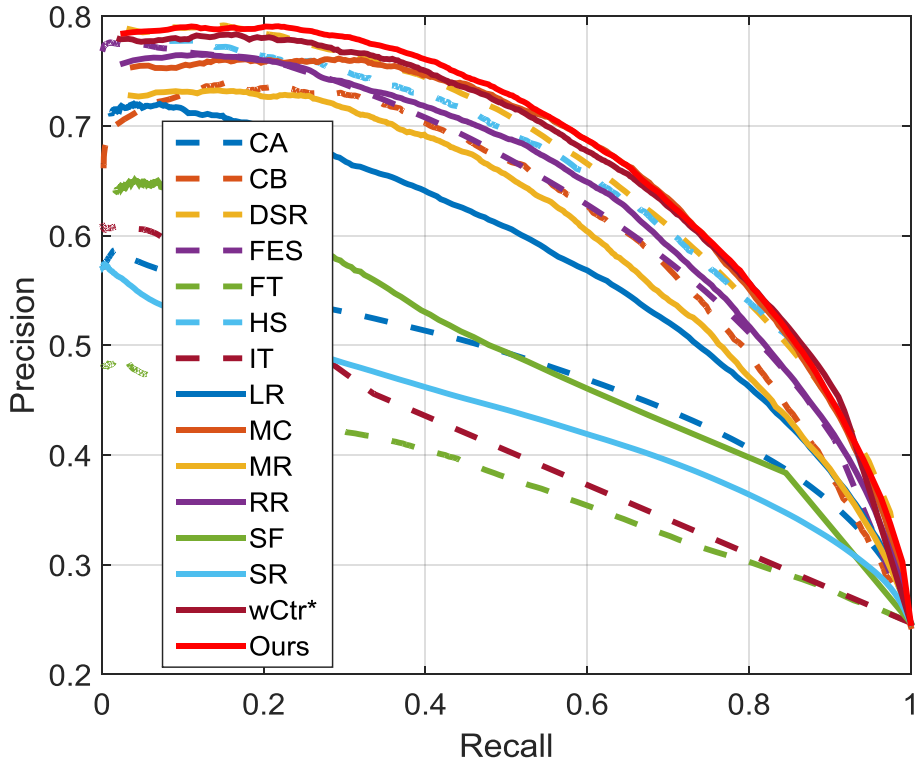

Figure 4.17 Precision-recall curves on the PASCAL-S dataset.

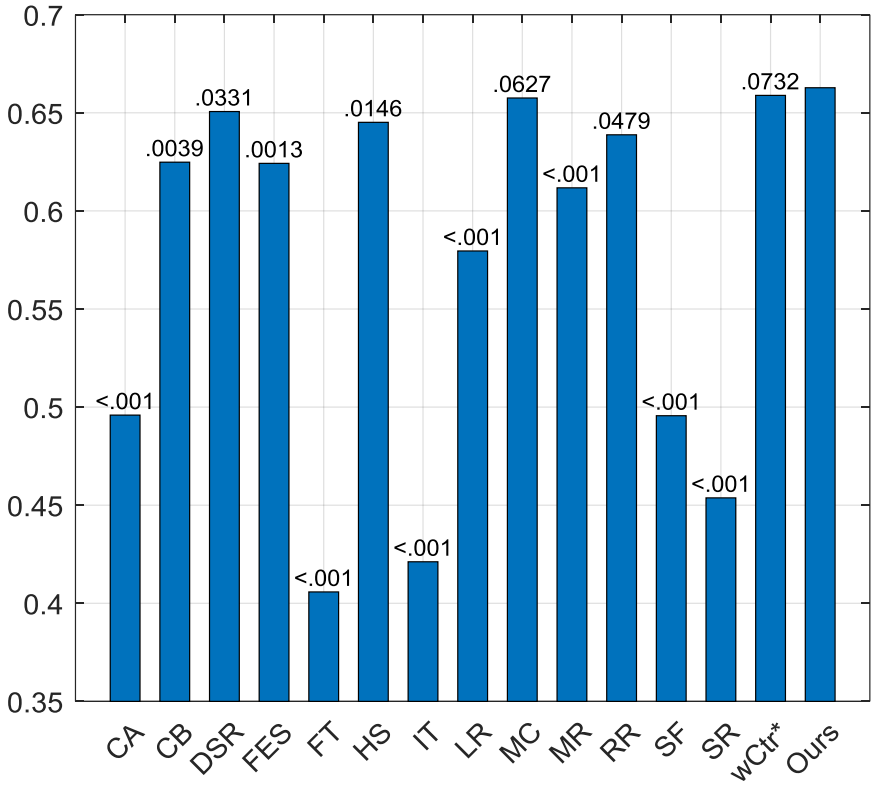

Figure 4.18 F-measures on the PASCAL-S dataset.

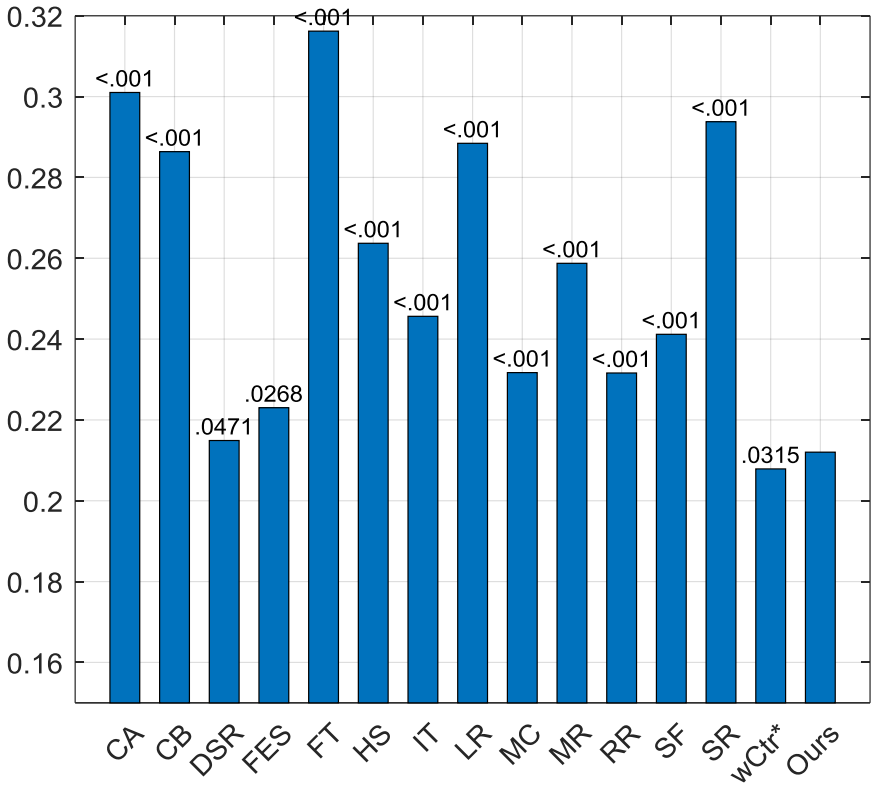

Figure 4.19 MAE scores on the PASCAL-S dataset.

Finally, we conduct evaluation on the newly proposed BAOS dataset. The results in Figure 4.20 to Figure 4.22 display the absolute advantage of our method. It not only has a significantly higher precision-recall curve in Figure 4.20, but also obtains the optimal F-measure (0.742) and MAE (0.296) in Figure 4.21 and Figure 4.22, with statistically significant advantages. The dominance of our method on the BAOS dataset demonstrates its elevated robustness to salient objects on the image boundaries.

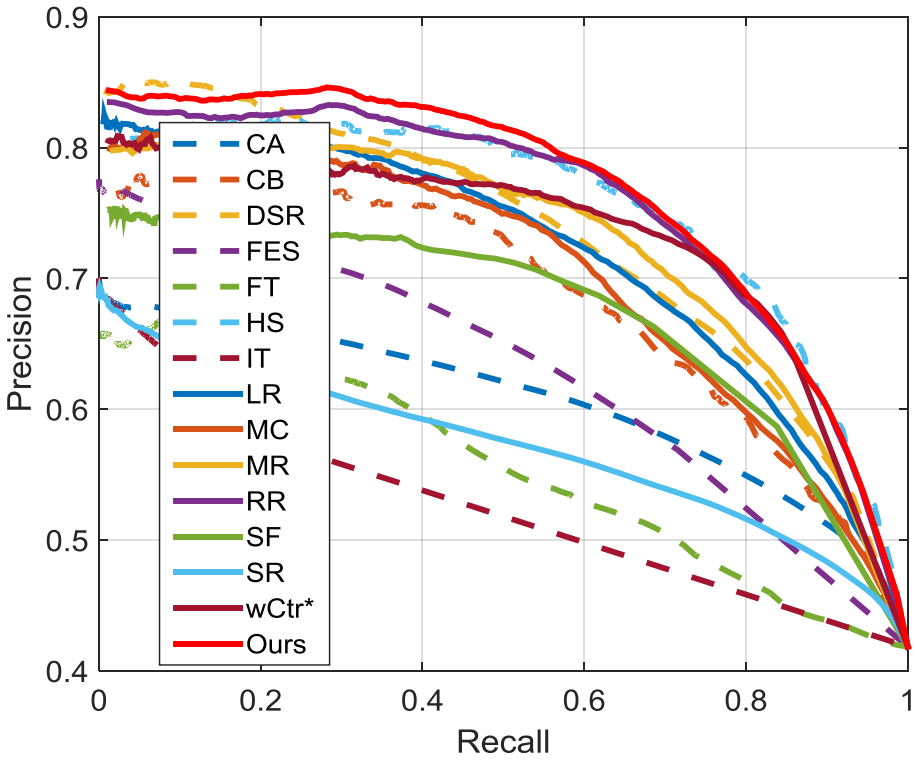

Figure 4.20 Precision-recall curves on the BAOS dataset.

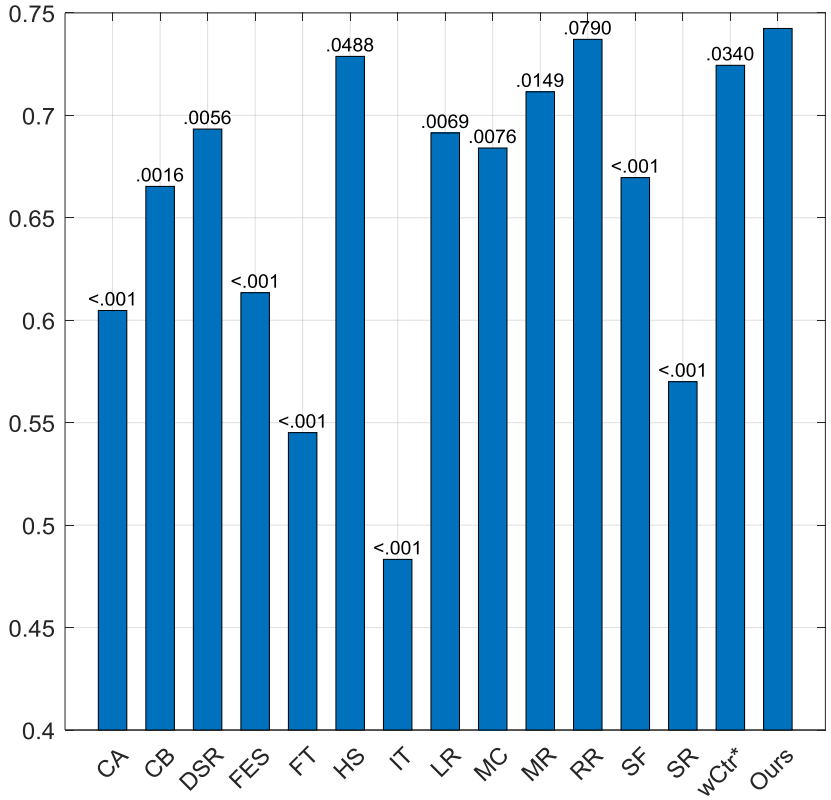

Figure 4.21 F-measures on the BAOS dataset.

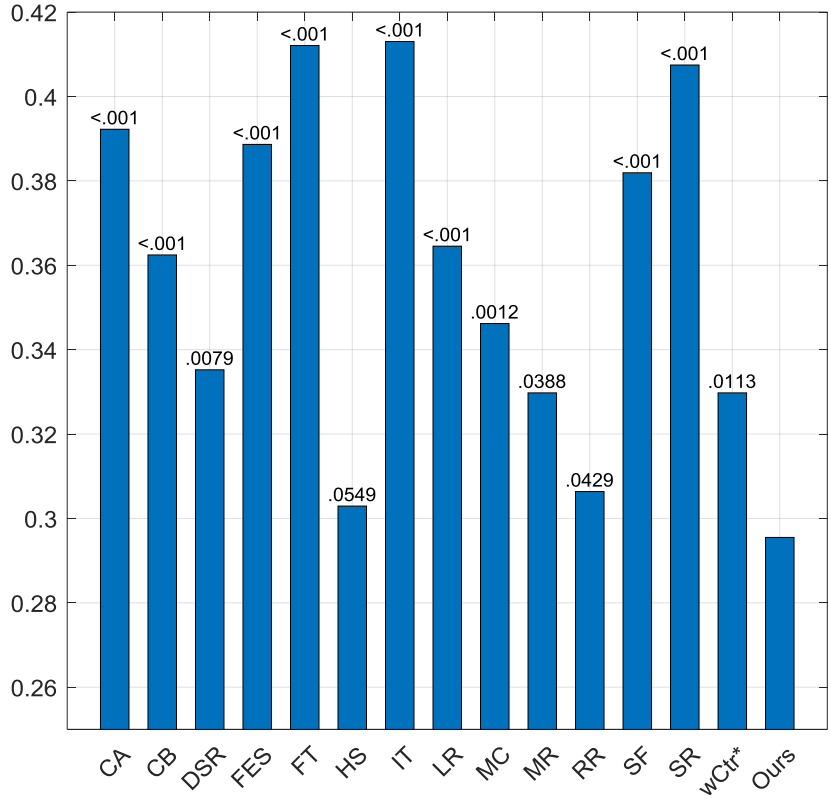

Figure 4.22 MAE scores on the BAOS dataset.

Qualitative Evaluation:

To provide a qualitative comparison of the different saliency outputs, we select eight example saliency maps from each of the fifteen methods, and tile them in Figure 4.23.

We select the top six methods in Table 4.3 with the best performances, namely DSR, HS, MC, MR, wCtr* and RR, in the qualitative evaluation against our proposed RCRR method. Through the visual examples in Figure 4.23, we observe that in general, RCRR achieves the best performance among the chosen images. The comparison methods are analyzed below.

(1) The DSR [80] method computes saliency via multi-scale reconstruction errors followed by an object-based Gaussian refinement. However, since the saliency map boundaries are frequently suppressed by the Gaussian refinement, DSR will always tend to produce dark boundaries, which is clearly visible on all of the chosen images.

(2) The HS [69] method is ideal in dealing with small-scale high-contrasts regions by the use of a tree model. Yet, since it depends on the extraction of cue maps with low-level features such as color and position, it does not work well with images that have low contrast between the foreground and the background, e.g. Figure 4.23(b) and (c).

(3) The MC [54] model applies absorbed time of Markov chain in calculating the saliency value, and provides fair enough estimations in most cases. Nevertheless, it tends to highlight the center due to its longer distance to the boundaries, and will frequently fail in detecting boundary-adjacent salient objects, which is seen in Figure 4.23(a), (b) and (h).

(4) The MR [52] method evaluates superpixel saliency via graph-based manifold ranking, which functions well in images with centered salient objects. However, it completely relies on the image boundaries as background queries, which greatly suffers

the aforementioned saliency reversion problem when boundary-adjacent salient objects are presented, as witnessed in Figure 4.23(a), (b), (e), (f) and (g).

(5) The wCtr* [53] method optimizes the saliency detection by exploiting the proportion that a region connects to the boundaries, which shows good results on centered salient objects. However, its core idea, the boundary connectivity, still uses image boundaries as background queries, which suffers similar drawbacks as the MR method does.

(6) Finally, the RR [1] method is a former version of RCRR, and instead of applying RC, it uses 3 of the 4 image boundaries as background queries. Figure 4.23(a), (b), (d) and (g) demonstrate that the proposed RC step can provide even higher robustness than the boundary selection strategy of RR.

On the other hand, our method generates saliency maps that visually correlate with the ground truth better. It exhibits high robustness under various cases, even in the cases with complex backgrounds such as in Figure 4.23(b) and (f). With the improvement from the proposed RC step, it shows marked advantage in handling boundary-adjacent salient object images, minimizing the emergence of saliency reversion. Moreover, the proposed RRWR step helps to provide elevated accuracy and smoothness to the output saliency map, which are seen in Figure 4.23(c), (d) and (h). We further note that our method is good at suppressing background regions that share similar patterns to the salient object, such as Figure 4.23(b).

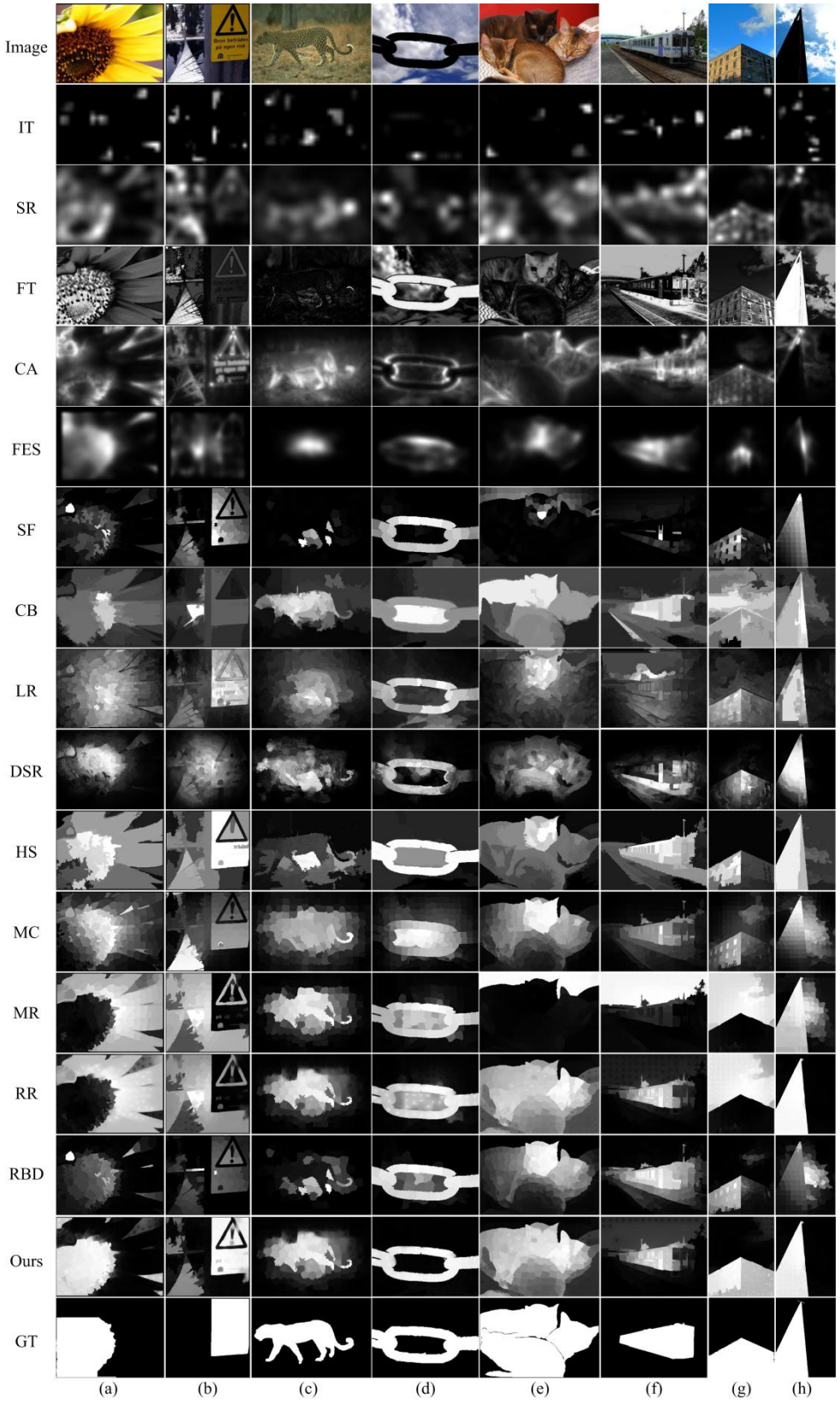

Figure 4.23 Saliency map examples of state-of-the-art methods against our RCRR method. (a) – (h): Image case IDs.

4.5.7 Extensibility as A Saliency Optimization Algorithm

As stated in section 4.4.1, the initial saliency estimation can also be generated by other boundary-based methods. In such a case, we suggest that our method, including the RC step and the RRWR step, functions as a saliency optimization algorithm. To evaluate its optimization performance, we compare our method with RBD [53], which is a state-of-the-art saliency optimization algorithm that can be widely applied on different saliency detection methods for performance improvements.

We select two boundary-based methods as the to-be-optimized methods, namely MR [52] and MC [54]. The same five datasets from section 4.5.6 are used. The results are listed in Figure 4.24 to Figure 4.27. It is obvious that our method outperforms RBD among all of the F-measure bars in Figure 4.24 and Figure 4.26. Our method also achieves the lowest MAE scores on all of the five datasets in Figure 4.25 and Figure 4.27, when compared to both the original methods and their RBD-optimized versions. The improvement of our method over RBD lies in the fact that RBD is reliant on image boundaries as the background queries, which inevitably suffers from the saliency reversion cases. It is also worth noting that our method shows especially high performance on the BAOS dataset, which further validates its high robustness on boundary-adjacent salient objects.

We note that RC and RRWR can also be independently exploited as two separate saliency optimization algorithms, which provides further flexibility of our method in practical applications.

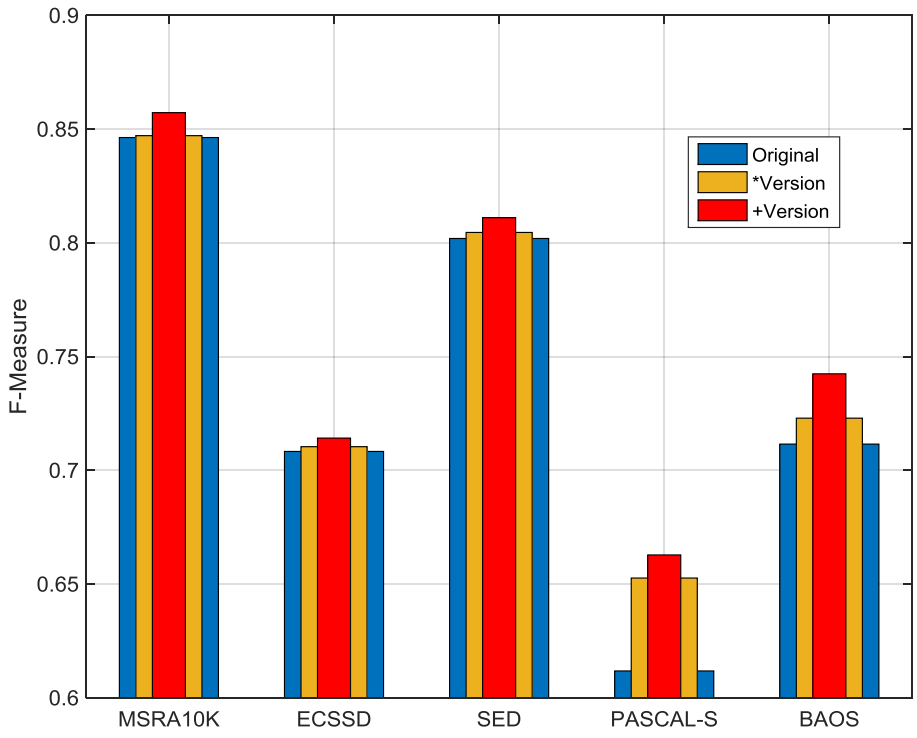

Figure 4.24 Optimization evaluation results of F-measure on the MR method.

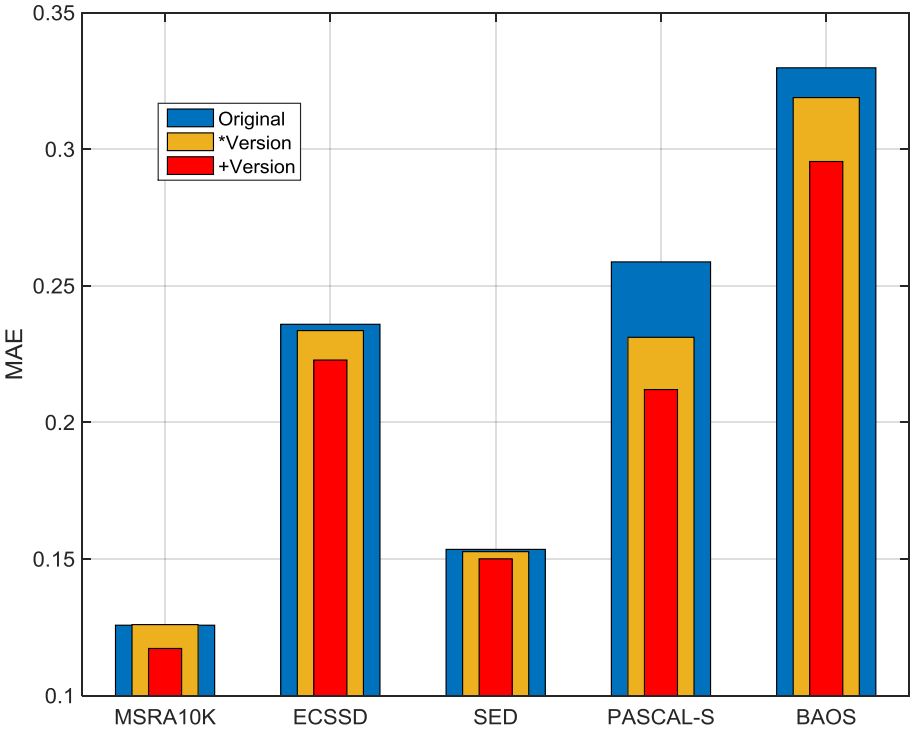

Figure 4.25 Optimization evaluation results of MAE on the MR method.

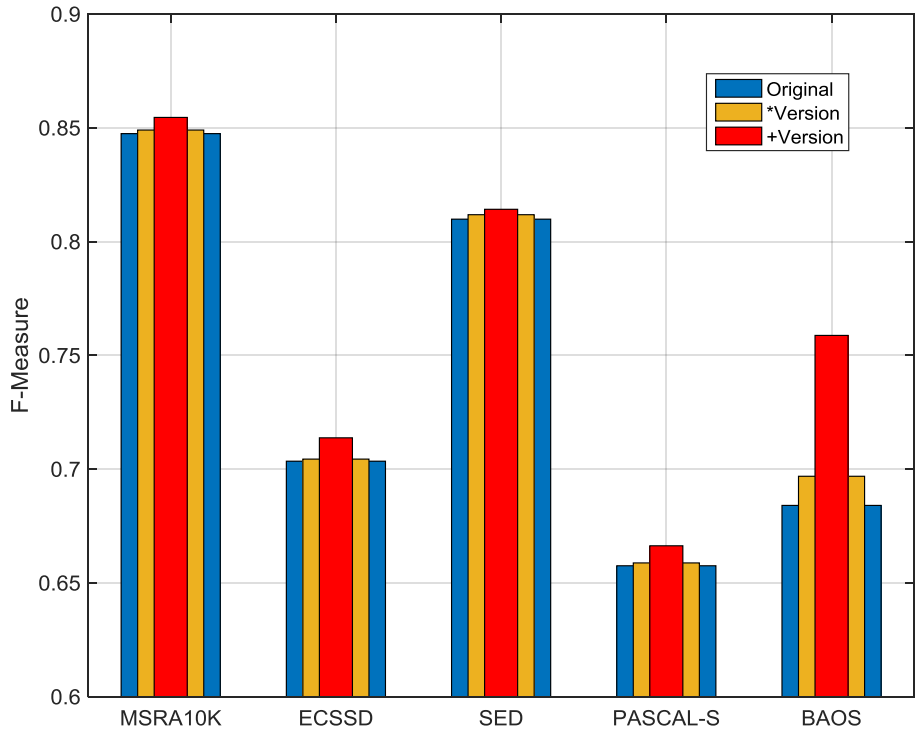

Figure 4.26 Optimization evaluation results of F-measure on the MC method.

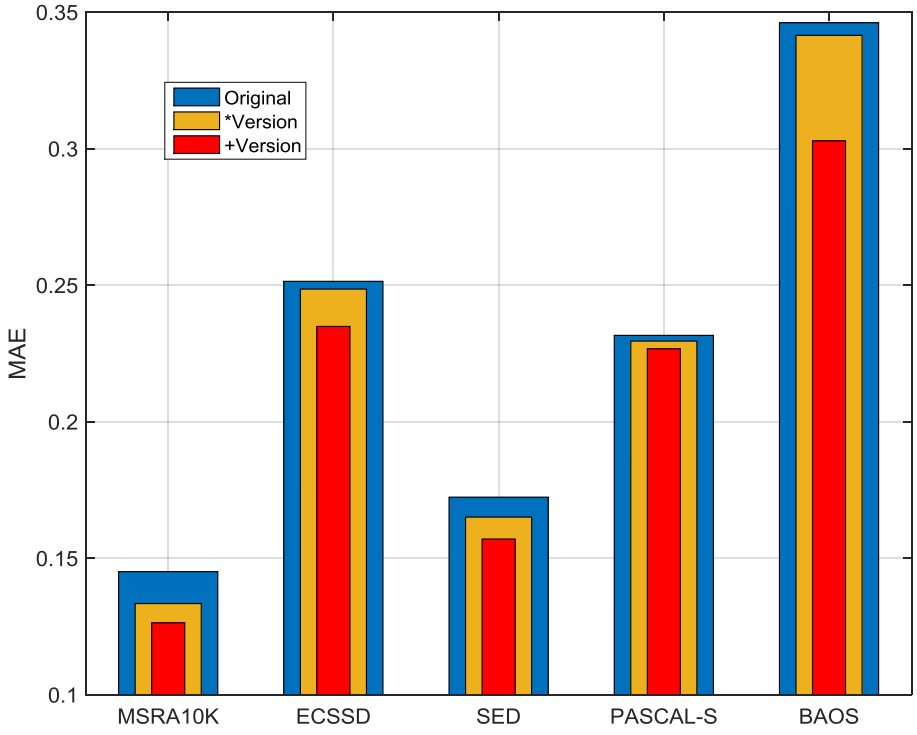

Figure 4.27 Optimization evaluation results of MAE on the MC method.

4.5.8 Efficiency

Our method is implemented on the machine described in section 4.5.4. The average calculation time per image of our method is 0.408s (excluding the time for superpixel generation and initial saliency estimation), in which the RC step takes less than 0.01s, and the RRWR step takes 0.358s.

4.5.9 Limitation

One limitation of RCRR, as observed in the experiments, is that in images where the salient object occupies more than half (or even all) of the image boundaries, the originally correct initial saliency estimation S_{init} will be mistakenly detected as "reversed", and thus be unnecessarily processed by the RC step, as the example in Figure 4.28 shows. Nevertheless, considering that such cases only appear occasionally, our method still prevails over the other state-of-the-art methods in the overall performance.

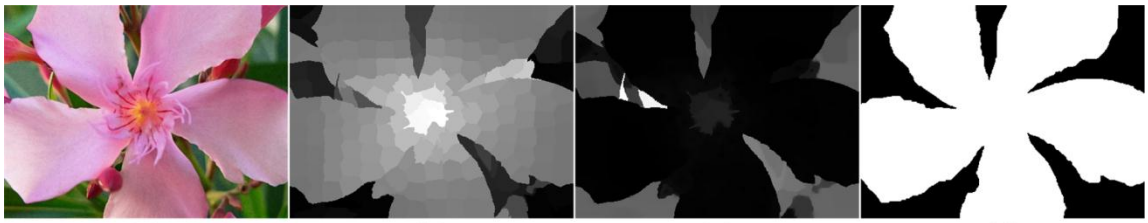

Figure 4.28 Example case showing the limitation of our proposed RCRR method. (a) Input image; (b) result of MR; (c) result of RCRR; (d) ground truth.

4.6 Summary

In this chapter, we have proposed RCRR, a novel saliency detection method based on improved low-level image features. The significant contributions of our method lie in two aspects: firstly, the RC step can effectively neutralize the negative influences of the

boundary-adjacent foreground regions, and thereby reversing the “reversed” saliency maps back to normal, leading to more accurate and robust saliency estimations; secondly, the RRWR step can provide prior saliency estimation to all of the pixels in an image, resulting in smoother and more detailed saliency map output. We also distribute the BAOS image dataset, which can be used to evaluate the performance on boundary-adjacent salient objects. Our method is fully automatic without any user supervision. Results of experiments on five datasets show that our method significantly outperforms fourteen state-of-the-art saliency detection methods in both accuracy and robustness, while maintaining relatively high efficiency. We further demonstrate the extensibility of our method as a saliency optimization algorithm.

Chapter 5 DNN Based Saliency Detection

In Chapter 3 and Chapter 4, we have introduced two saliency detection methods based on low-level image features. These methods perform well against other state-of-the-art methods that also based on low-level features. However, the absence of high-level feature extraction makes them particularly vulnerable when encountering low contrast images and complex patterned images, as seen in section 1.3.3. On the other hand, the recently prevalent deep neural networks (DNNs), especially the convolutional neural networks (CNNs), are proved to be of great value in high-level feature extraction for saliency detection.

In this chapter, we propose two DNN based methods to further improve our saliency detection performance, namely the adaptive background search and foreground estimation (BSFE) method, and the dense and sparse labeling (DSL) method. Section 5.1 summarizes the challenges we are going to address; section 5.2 lists our contributions and the major steps in the two methods we propose; section 5.3 reviews the related works to our DNN based methods, namely auto-encoder, DNN based sparse labeling, and DNN based dense labeling; section 5.4 and 5.5 give step-by-step methodology of BSFE as well as its experimental results; section 5.6 and 5.7 give step-by-step methodology of DSL as well as its experimental results; and finally, section 5.8 concludes this chapter.

5.1 Problem Formulation

As introduced in section 2.1, conventional low-level image feature based saliency detection methods have shown promising results in both bottom-up methods and top-down methods. Nevertheless, at least three major drawbacks hinder the performances of these methods.

(1) In general, without feature abstraction and learning, the hand-crafted low-level features are only effective on relatively high contrast images and do not perform well on images with complex foreground / background contexts. This drawback, however, can be readily solved via high-level feature learning, which is seen in Figure 5.1(a).

(2) Most of the prior knowledge applied in low-level feature based methods is largely empirical with specific pre-assumptions, e.g. image boundary regions are assumed as background [52], [53], or image center regions are assumed as foreground [54], [55]. These pre-assumptions are easily violated on broader datasets with more unusual-patterned images, as in the example in Figure 5.1(b). This issue has been discussed in chapter 3 and chapter 4, and remarkable improvements have been presented. Yet, these boundary refinement processes (such as the RC step in chapter 4) are still restricted in empirical pre-assumptions of image boundaries, and a high-level image feature based approach is desired to provide prior knowledge for saliency estimations.

(3) Each low-level feature is usually advantageous only in a specific aspect, e.g. color histogram is good at differentiating texture patterns, while frequency spectrum is good at differentiating energy patterns. It is generally difficult to combine different low-level features into a single algorithm to benefit from them all. Although some integration

trials have been made [57], [58], these specially designed algorithms are bulky and inefficient due to the large number of features involved.

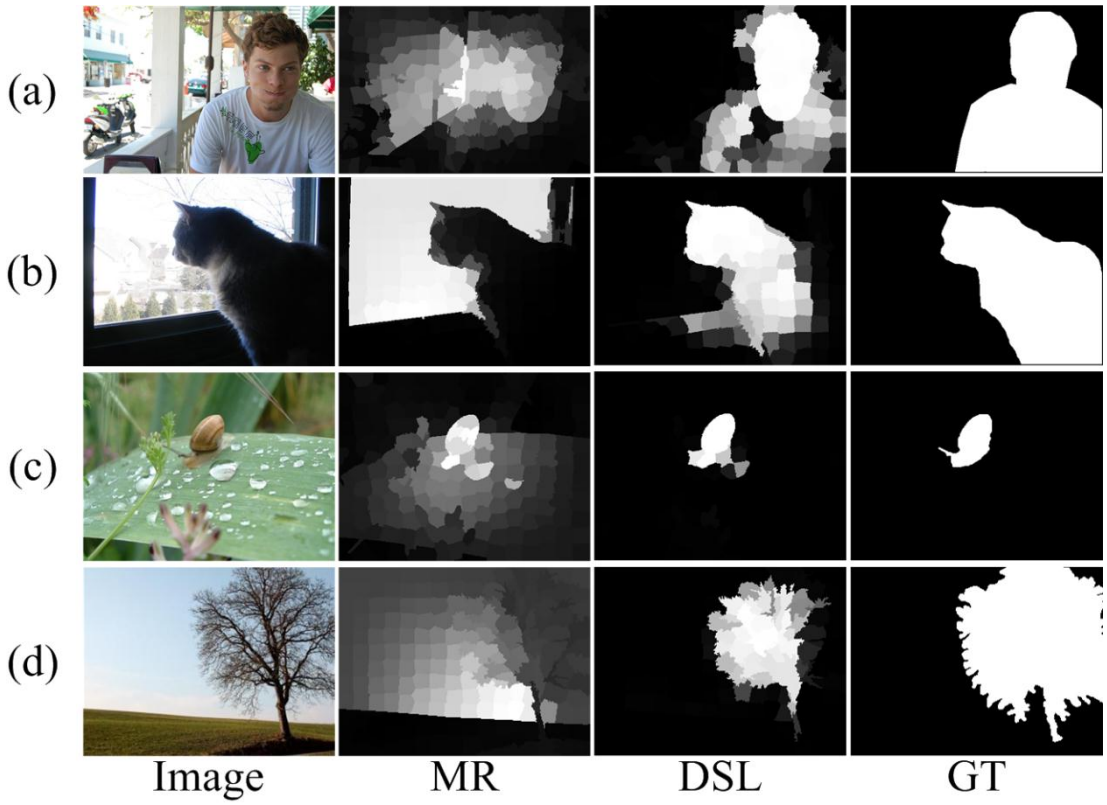

Figure 5.1 Illustration of challenges encountered by conventional low-level feature based saliency detection methods. (a) – (d): Image case IDs. From left to right: input images, saliency maps by a low-level feature based method [52], saliency maps by our proposed DSL method, ground truth.

5.2 Contributions

In this chapter, to address the three issues above faced by conventional low-level feature based methods, we propose BSFE and DSL, which are two DNN-based methods for saliency detection. The key contributions of these two methods are listed below.

For BSFE:

(1) We propose an adaptive background extractor, which approximates background regions semantically and cognitively, contributing to higher detection accuracy;

(2) We apply the auto-encoder (AE) hierarchically for foreground estimation, which is guided by the background mask, to reconstruct the final saliency map with higher performance.

And for DSL:

(1) We combine the DNN-based dense labeling (DL) and sparse labeling (SL) together for initial saliency estimation, in which DL conducts dense labeling that maximally preserves the global image information and provides accurate location estimation of the salient object, while SL conducts sparse labeling that focuses more on local features of the salient object;

(2) For the SL step, both low-level features and RGB features of the image are applied as the network inputs. Such multi-dimensional input features enable the complementary advantage of low-level features and RGB features, by which the image is more accurately abstracted and represented;

(3) In the last deep convolution (DC) step, a 6-channeled input structure is proposed, which provides significantly better guidance in generating the final saliency map. On the one hand, the combined initial saliency estimations from the DL and SL steps provide accurate location guidance of the salient object, effectively excluding any false salient region (Figure 5.1(c)); on the other hand, the superpixel indication channel precisely represents the current to-be-classified superpixel, which leads to more consistent and accurate saliency labeling (Figure 5.1(d)).

Both of the proposed methods are evaluated on publically available datasets, where BSFE is evaluated on four datasets against six state-of-the-art methods, and DSL is evaluated on six datasets against sixteen state-of-the-art methods (including ten

conventional methods and six learning based methods). Both methods have shown their superior performances in the experimental results.

5.3 Related Works

Section 2.4 has already introduced the fundamentals of DNN, as well as its sparse and dense labeling applications. In this section, we briefly review the basic principles and applications of auto-encoder (AE) as supplementary preliminary knowledge.

5.3.1 Auto-Encoder

Auto-encoder (AE) is one of the simplest forms of neural networks. It aims to convert the network input data into outputs with the least amount of distortion by learning patterns from the input data [175].

Classical AE is an unsupervised learning algorithm that applies back-propagation and makes the target values of the network outputs equal to the inputs [176].Specifically, it consists of an encoding process and a decoding process. The encoding process takes an encoding function $f(x_i, \theta_f)$ (usually the sigmoid function $sig(x) = 1/(1+\exp(-x))$) to make the transformation

$$y_i = f(x_i, \theta_f) = sig(Wx_i + b) \tag{5.1}$$

where y_i is the output of the hidden layer, $\theta_f = \{W, b\}$, W is a projection matrix, and b is a bias term. On the other hand, the decoding process adopts a decoding function $g(y_i, \theta_g)$ to map the hidden representation y_i to a reconstruction representation z_i:

$$z_i = g(x_i, \theta_g) = sig(W' y_i + b') \tag{5.2}$$

where $\theta_g = \{W', b'\}$. After the decoding process, z_i is taken as the prediction of the input x_i.

The training of an AE is to optimize the parameters $\theta_f = \{W, b\}$ and $g(y_i, \theta_g)$ so that the mean-squared error between the training data and their predictions is minimized:

$$\arg\min_{\theta_f, \theta_g} L(X, Z) \tag{5.3}$$

$$L(X, Z) = \frac{1}{2}\sum_{i=1}^{m} \|x_i - z_i\|^2 \tag{5.4}$$

where $X = \{x_i\}, Z = \{z_i\}, i = 1, 2, ..., \mathrm{m}$.

In practice, the stacked auto-encoder (SAE) is more prevalently used. An SAE is comprised of multiple unsupervised feature learning layers, which can be trained via greedy methods for each additional layer. To be specific, once the first layer is trained, its output will become the input of the second layer, and all the additional layers will be trained this way. The deep architecture of SAE grants it the ability to learn more complex and abstract features during training.

5.4 Saliency Detection with Adaptive Background Search and Foreground Estimation (BSFE) Using Comprehensive Auto-Encoder

Our proposed BSFE consists of two individual SAEs, one for the adaptive background search (BS), and the other one for the foreground estimation (FE).

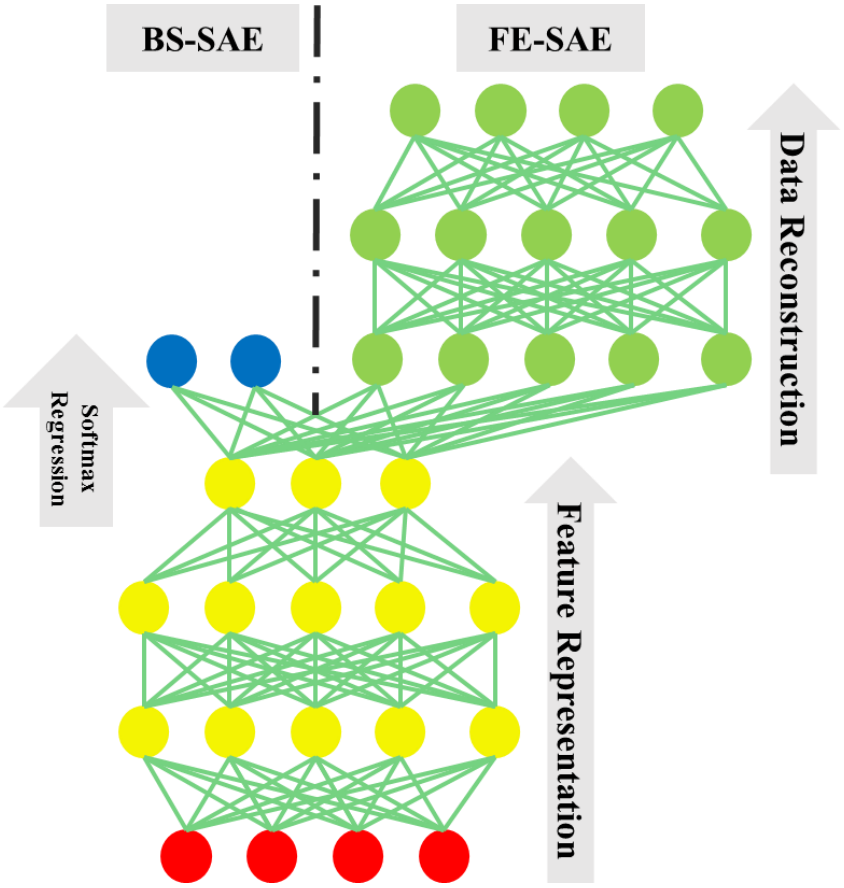

Figure 5.2 Flowchart of our proposed BSFE method.

5.4.1 Adaptive Background Search

We first extract a rough background estimation of an image by our proposed BS SAE model. Specifically, for an RGB image patch p_{bs} with the size of $m \times m$ pixels from the training image I, the input vector $f(p_{bs})$ of BS SAE is obtained by

$$f(p_{bs}) = \begin{bmatrix} g(p_{bs}) \\ g(\hat{I}) \end{bmatrix} \tag{5.5}$$

where $\hat{I} \in \mathbb{R}^{m \times m \times 3}$ is the resized image of I, and following [90], m is set to 51; $g(\cdot)$ is the vectorization operation, and thus $f(p_{bs}) \in \mathbb{R}^{15606 \times 1}$. As $f(p_{bs})$ is the concatenation of

local context (p_{bs}) and global context ($\hat{I}$), the trained BS SAE model can infer background region from a holistic view, rather than be restricted to local view [90] or regional view [177].

After obtaining the feature representations of an image patch by the trained BS SAE model, we use the softmax function to measure its probability of being background. This grants us a background mask M_{bs} of I, which can be utilized for foreground estimation in section 5.4.2. As shown in Figure 5.3, compared with conventional boundary-background priors [1], [52], [54], [87], [175], [178], [179], the adaptive background mask M_{bs} is able to capture the background region semantically and cognitively.

Figure 5.3 Examples of the background mask by the BS SAE model.

5.4.2 Foreground Estimation

In the last section, we have generated the background mask M_{bs}. To improve the efficiency of our method, we transform M_{bs} to a superpixel-wise background mask and use superpixel as the unit for further operations. We partition each image into 250 superpixels using the SLIC algorithm [166]. The superpixel-wise background mask is

achieved by calculating the mean value of pixels within each superpixel. For brevity, we still use M_{bs} to denote the superpixel-wise background mask, unless otherwise specified.

With the testing image I and the corresponding background mask M_{bs}, we then construct the foreground estimation SAE model (FE SAE) to extract the foreground saliency of I. Different from BS SAE, the RGB histogram of the superpixel (with 20 bins in each color channel) is exploited as the input vector, and there is no softmax regression in FE SAE, which makes it a completely unsupervised learning model. Only the superpixels on M_{bs} with values more than 0.7 are selected as the training set for the FE SAE model.

After the training of FE SAE, we calculate the reconstruction residual $r_{p_{fe}}$ for each superpixel p_{fe} of I by

$$r_{p_{fe}} = \left\| h(p_{fe}) - \bar{h}(p_{fe}) \right\| \tag{5.6}$$

where $h(p_{fe})$ is the original input vector corresponding to p_{fe} and $\bar{h}(p_{fe})$ is the data reconstruction of $h(p_{fe})$ by FE SAE. Inspired by [175], the idea of our method is that as FE SAE is constructed by the background superpixels, the superpixels belonging to background have low reconstruction residual, while the superpixels belonging to foreground have high reconstruction residual. The reconstruction residual is thus adopted to measure the saliency value of p_{fe} with the following formula:

$$\begin{cases} s_{p_{fe}} = \dfrac{1}{1+\exp(\dfrac{\xi(u-r_{p_{fe}})}{u-v})} \\ u = \max(\mathrm{r}_p : \mathrm{p} \in \mathrm{D}) \\ v = \dfrac{1}{|D|}\sum_{p \in D} r_p \end{cases} \tag{5.7}$$

where ξ is the smooth factor (which is set to 6 empirically); r_p is the reconstruction residual of superpixel p by (5.6); and D is the training set of FE SAE.

Considering that complex background may impede the accuracy of the foreground estimation, we hierarchically conduct the foreground estimation algorithm in regional scales for better performance. Specifically, the testing image I is first segmented into two regions by the Ncut algorithm [180]. Two individual FE SAEs are then constructed respectively under these two regions and each superpixel of I is assigned to the saliency value by (5.7) with the corresponding FE SAE. In the next hierarchy, we segment the two regions respectively to generate four smaller regions and construct four individual FE SAEs corresponding to these regions. Each superpixel of I is assigned to the new saliency value by (5.7) in this hierarchy. Note that in each segmentation operation, only two sub-regions are generated and the region is no longer segmented when $|D'| \le 0.3 \times |A|$ or $|D'| \ge 0.7 \times |A|$, where D' and A are the training set and superpixel set respectively corresponding to the region. This process is repeated until no more regions to be segmented. Finally, the saliency value of the superpixel is obtained by linearly combining the saliency values of each hierarchy.

The complete flowchart of the BSFE method is shown in Figure 5.4, and the foreground estimation algorithm is summarized in Table 5.1.

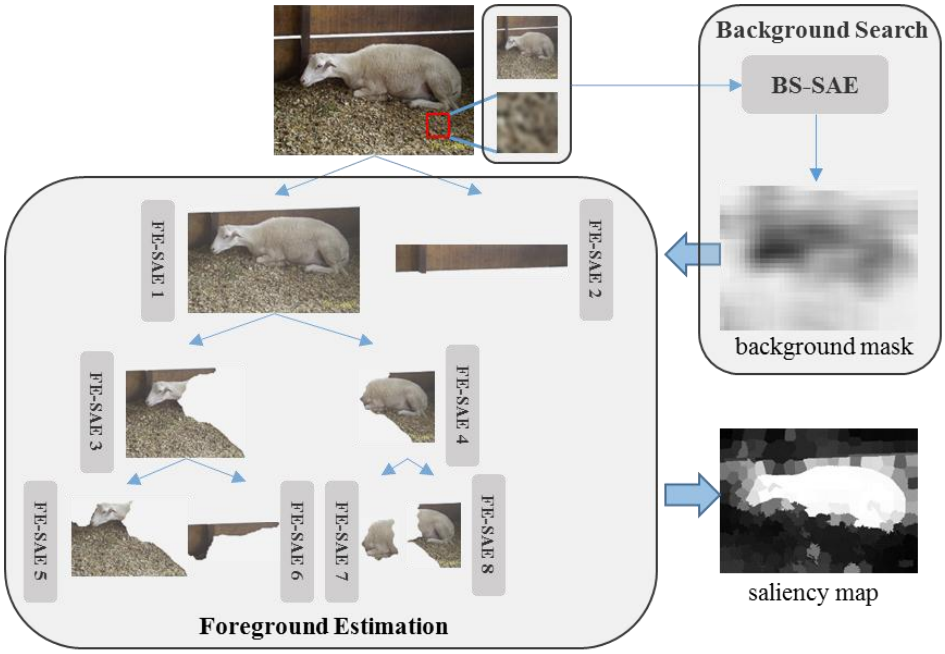

Figure 5.4 Flowchart of our proposed BSFE method.

Table 5.1 Algorithm description of our proposed foreground estimation

Step	Content
Input	Input image I and background mask M_{bs}.
1	$S=\{s_p\} \leftarrow 1-M_{bs}$
2	Segment I into two regions I_1 and I_2 by Ncut [180].
3	$O \leftarrow \{I_1, I_2\}$
4	**while** $O \neq \varnothing$:
5	**for each** $R \in O$:
6	select training set D'_R according to M_{bs}
7	train FE SAE
8	**for each** superpixel $p \in R$:
9	calculate saliency value s'_p by (5.7)
10	$s_p \leftarrow (s_p + s'_p)/2$
11	**end for**
12	remove R from O
13	**if** $0.3\times\lvert R\rvert \le \lvert D'_R\rvert \le 0.7\times\lvert R\rvert$ **then**:
14	segment R into R_1 and R_2 by Ncut
15	$O \leftarrow O \cup \{R_1, R_2\}$
16	**end if**
17	**end for**
18	**end while**
Output	Saliency map $S=\{s_p\}$.

5.5 Experimental Results of BSFE

In this section, the experimental results of our proposed BSFE method are presented. We first introduce the necessary setup of our experiments, including datasets, evaluation metrics and parameter assignments. And then we exhibit the comparison experiment of BSFE against six state-of-the-art methods.

5.5.1 Datasets

We select the MSRA10K [64] dataset for training, which contains 10,000 natural images with large variety and the corresponding pixel-wise saliency annotations. We randomly select 9,000 images from the dataset to train the BS SAE, and use the remaining 1,000 images for validation.

In testing, we adopt four public benchmark datasets, namely ECSSD [69], PASCAL-S [172], SED1 [171] and SED2 [171].

5.5.2 Evaluation Metrics

Following section 4.5.2, the precision-recall curve, F-measure and MAE score are also used in our experiments as the evaluation metrics.

5.5.3 Parameters

For the BS SAE model, we stack three AEs to extract feature representation in high-level manners, with 7,000, 3,500 and 2,000 hidden nodes in each AE, respectively. As suggested in [175], [177], before fed into BS-SAE, $f(p_{bs})$ is corrupted to enhance the robustness across a large training set, in which some of the units are set to be zero randomly.

For the FE SAE model, we stacked two AEs to boost the performance of data reconstruction, with 60 hidden nodes in each of the AE. As the number of training samples is small (generally less than 250), we did not corrupt the original input vector in FE SAE to make the trained model more specific to the small training set.

The hyper-parameters for the training of BS SAE and FE SAE are listed in Table 5.2.

Table 5.2 Hyper-parameters for the training of BS SAE and FE SAE

	BS SAE		**FE SAE**	
	Pre-training	**Fine-tuning**	**Pre-training**	**Fine-tuning**
Training epoch	15	60	15	100
Learning rate	1e-2	1e-6 for first 20 epochs; 8e-8 for next 40 epochs.	1e-2	1e-3

5.5.4 Implementation

Both BF SAE and FE SAE are implemented with the Theano frame [181], [182]. The machine used for our experiments is a PC with Intel 6-Core i7-5820K 3.3GHz CPU, 64GB RAM, GeForce GTX TITAN X 12GB GPU, and 64-bit Ubuntu 14.04.3 LTS.

5.5.5 Evaluation Against State-of-the-Art

Six popular state-of-the-art saliency detection methods are chosen as comparison methods against our proposed BSFE methods, which includes FT [66], LR [77], HS [69], MC [54], MR [52] and RR [1].

The quantitative experimental results are shown in Figure 5.5 to Figure 5.16. They demonstrate the superiority of our method on most datasets. Note that our BSFE method even achieved double-best results in terms of both FM and MAE on the PASCAL-S and

SED2 datasets, which are two of the datasets with more challenging scenarios and complex image patterns.

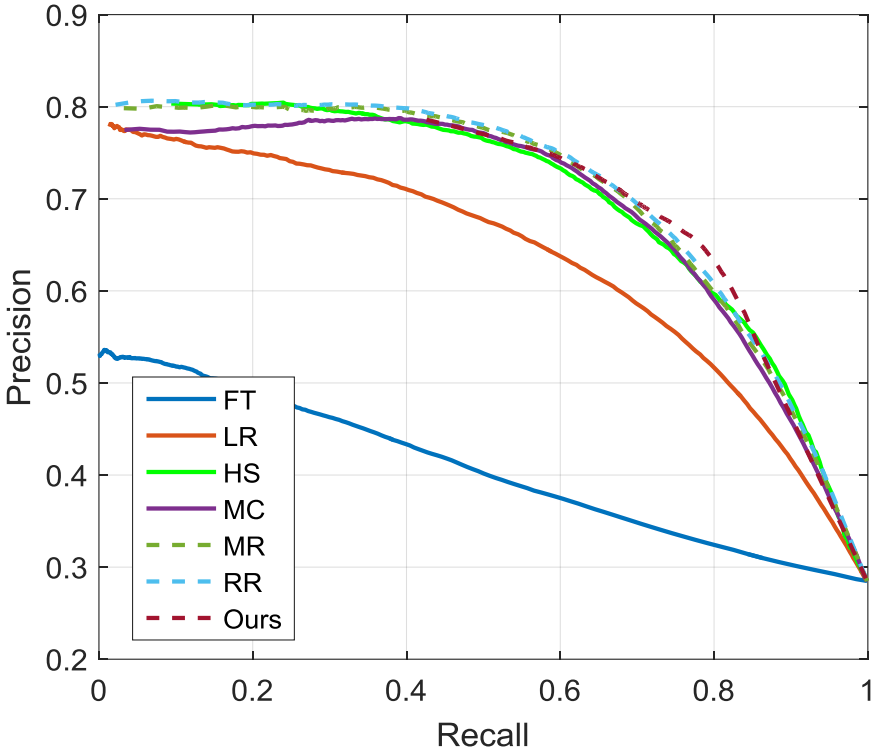

Figure 5.5 Precision-recall curves on the ECSSD dataset.

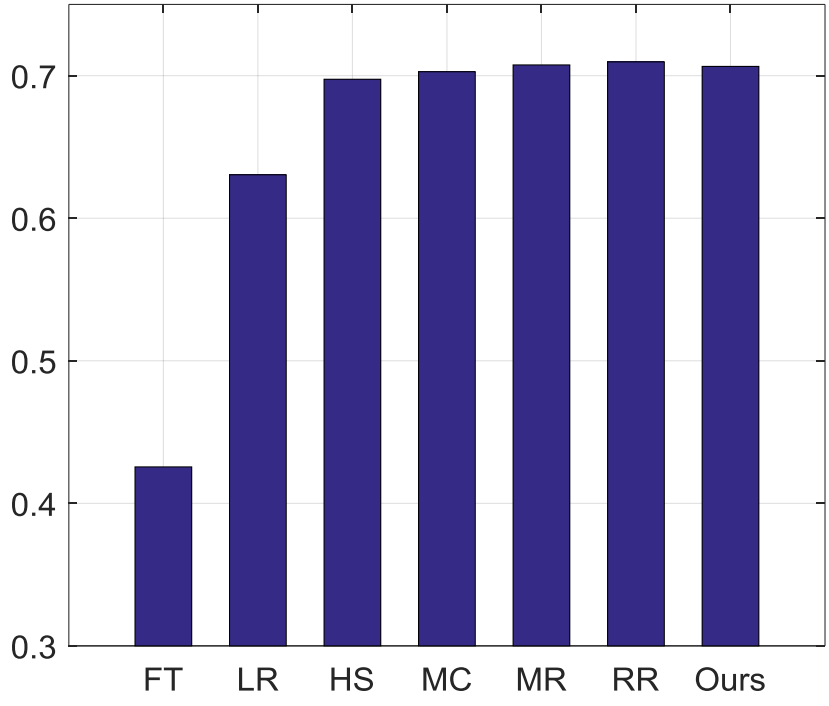

Figure 5.6 F-measures on the ECSSD dataset.

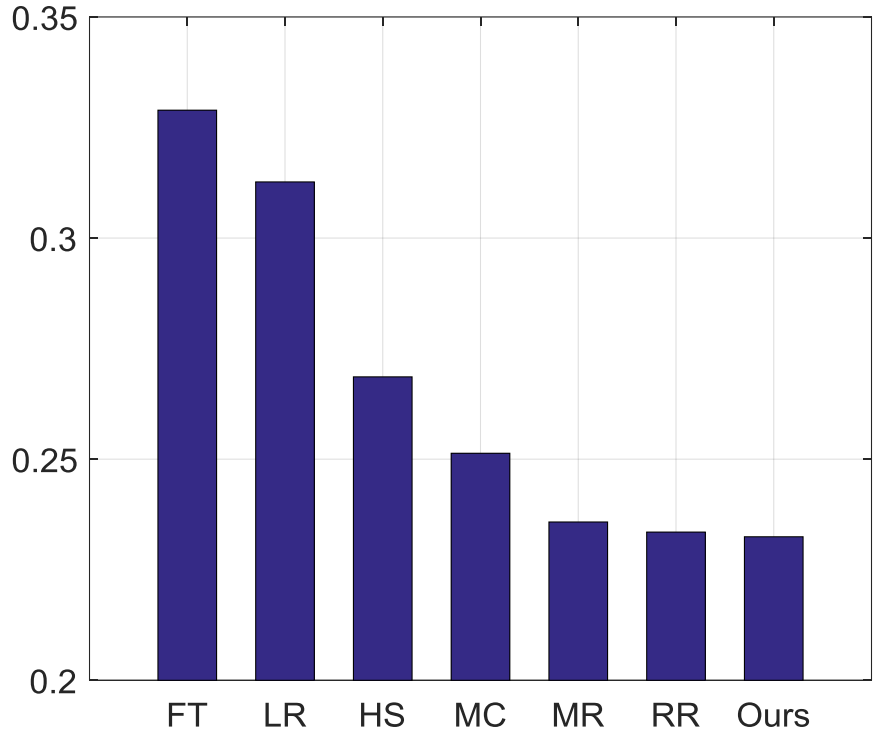

Figure 5.7 MAE scores on the ECSSD dataset.

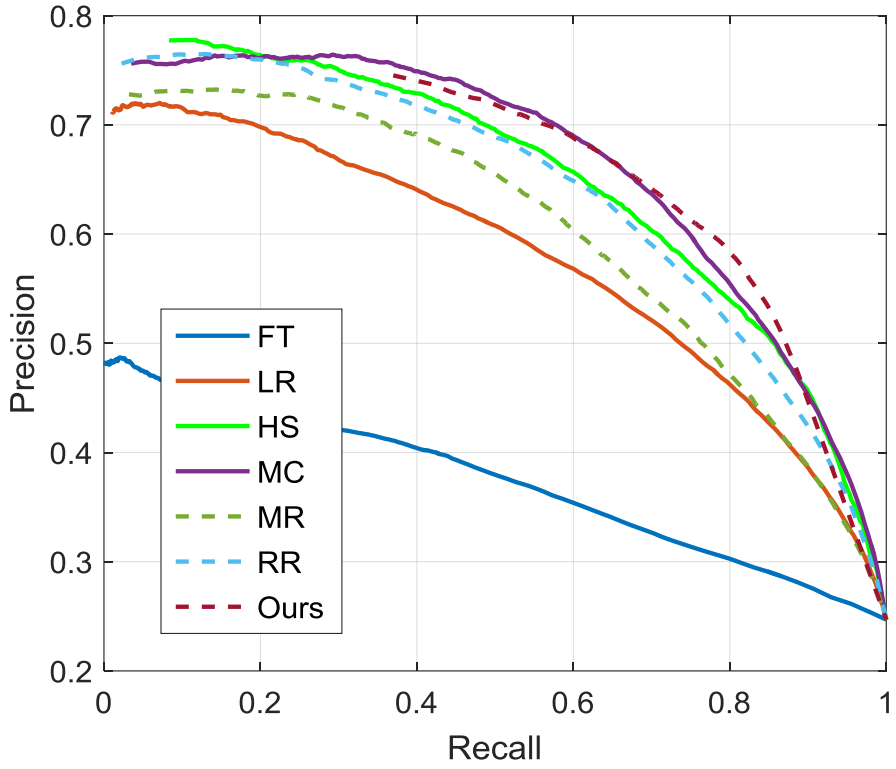

Figure 5.8 Precision-recall curves on the PASCAL-S dataset.

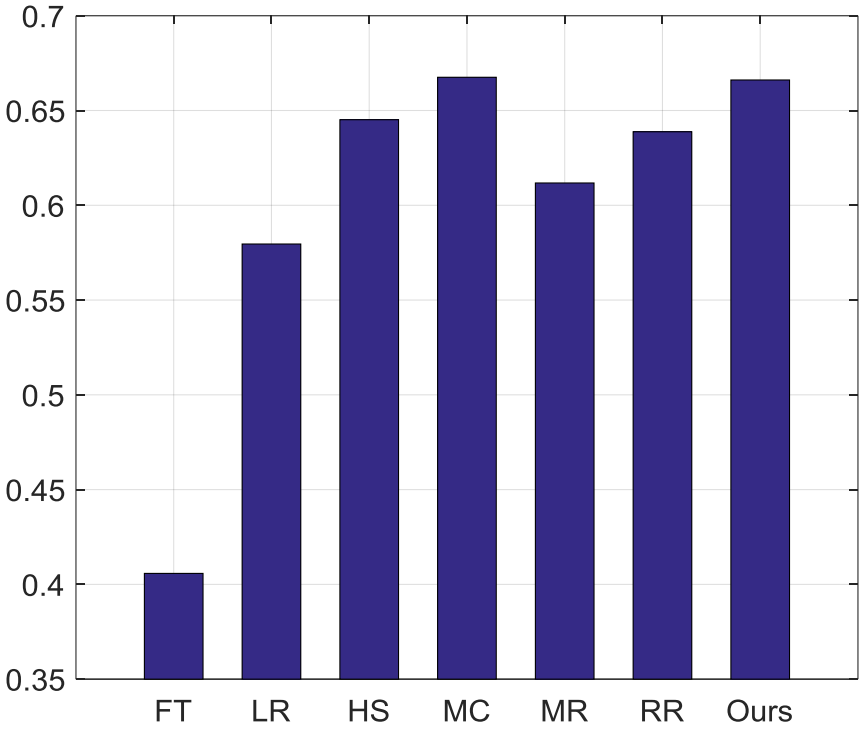

Figure 5.9 F-measures on the PASCAL-S dataset.

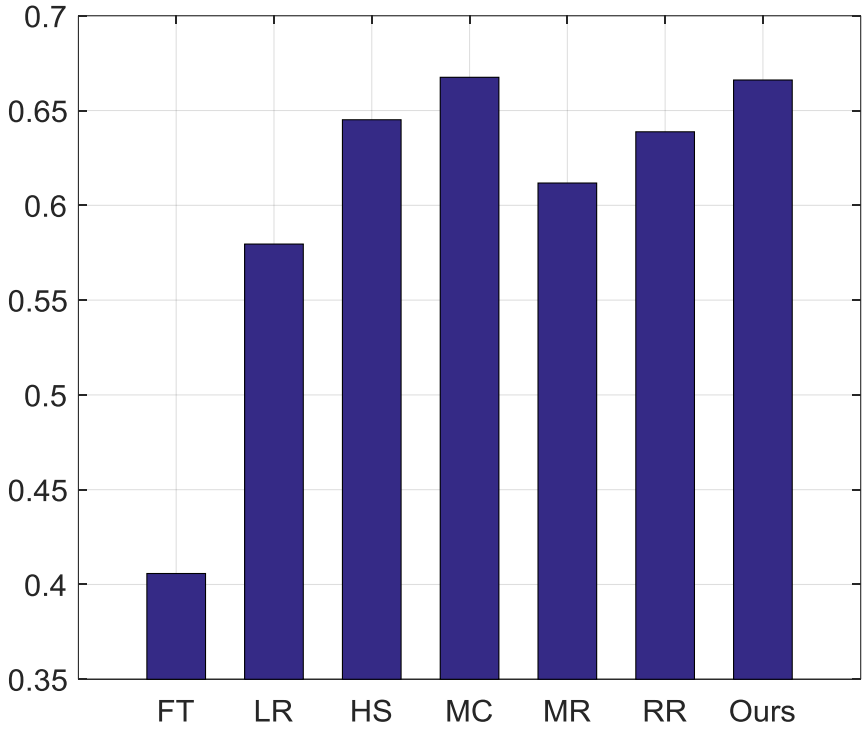

Figure 5.10 MAE scores on the PASCAL-S dataset.

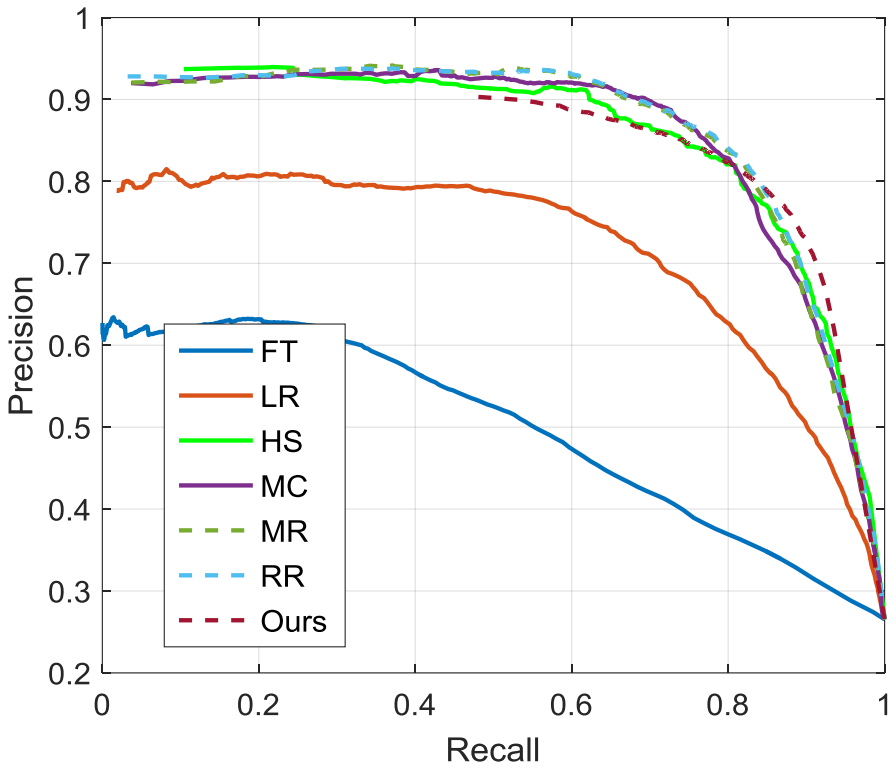

Figure 5.11 Precision-recall curves on the SED1 dataset.

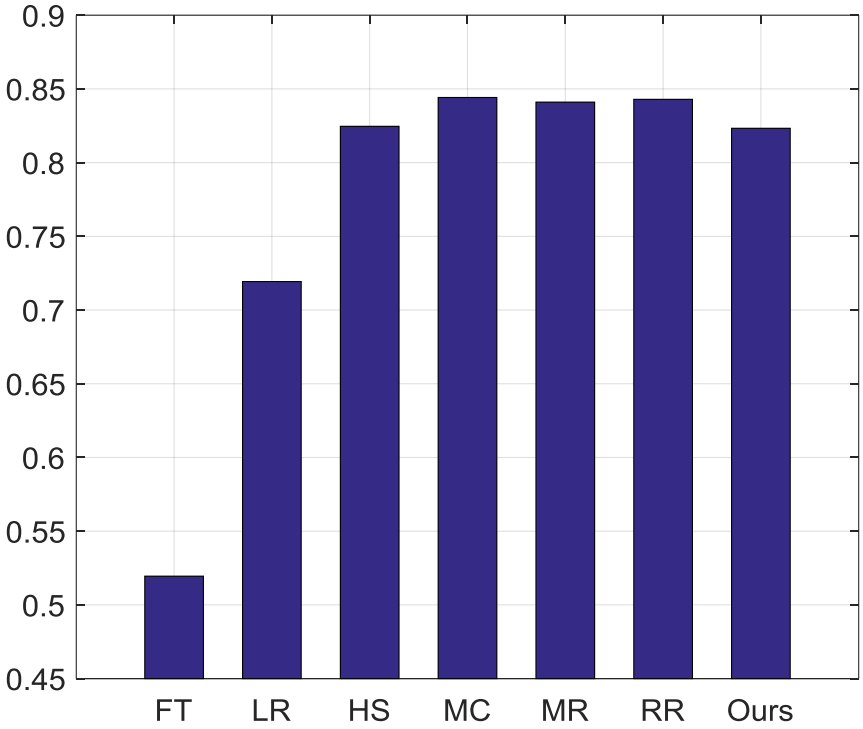

Figure 5.12 F-measures on the SED1 dataset.

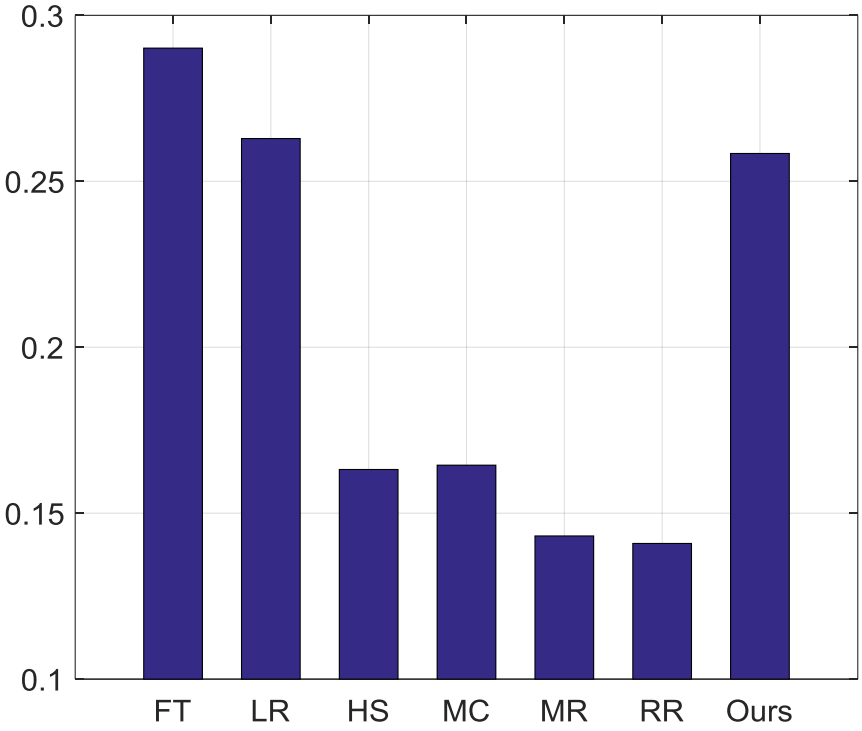

Figure 5.13 MAE scores on the SED1 dataset.

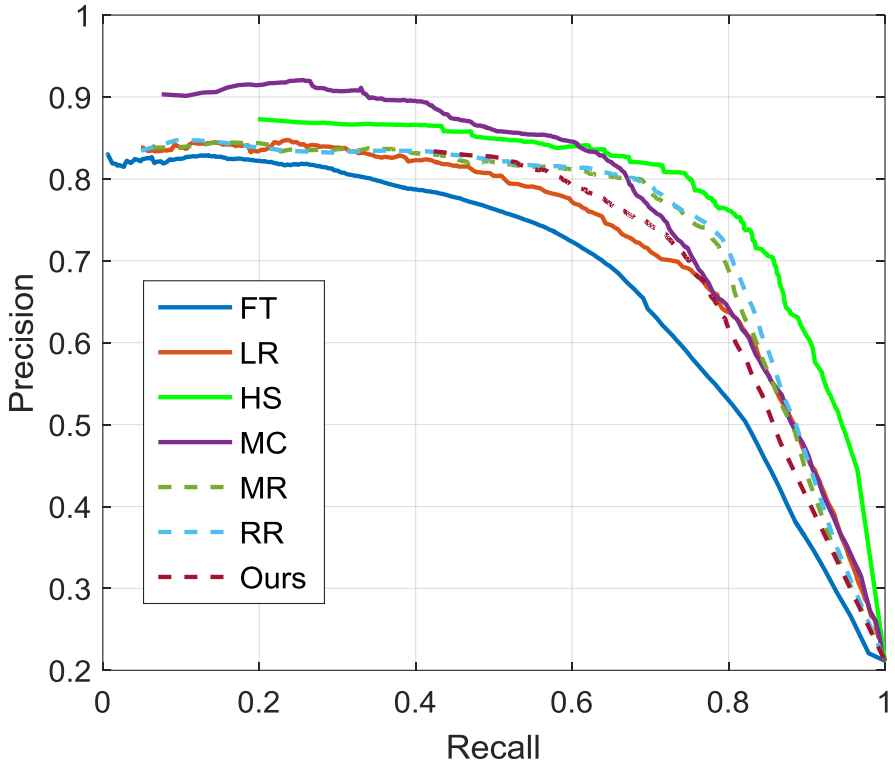

Figure 5.14 Precision-recall curves on the SED2 dataset.

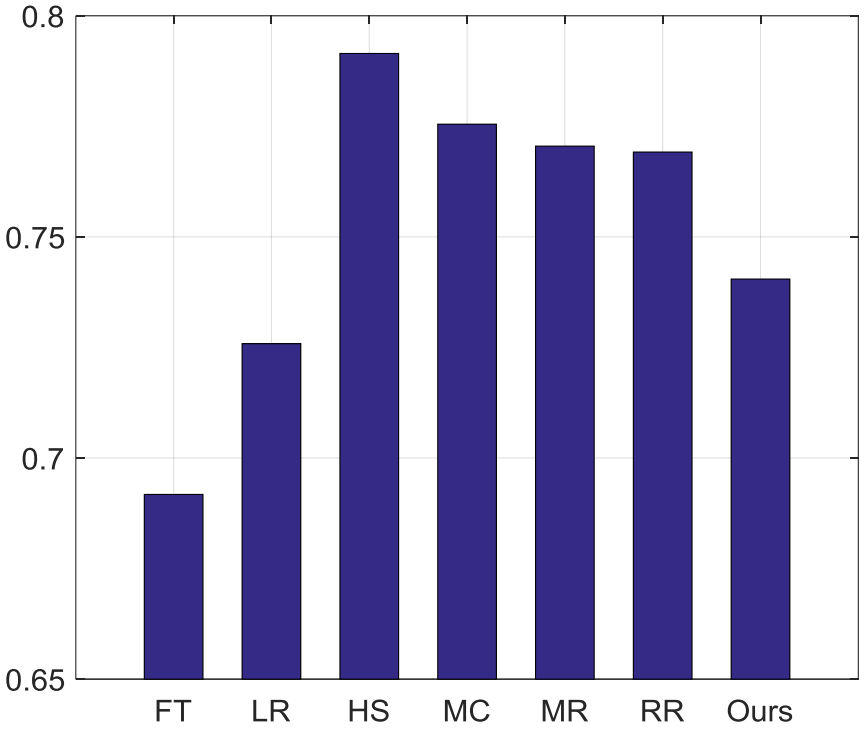

Figure 5.15 F-measures on the SED2 dataset.

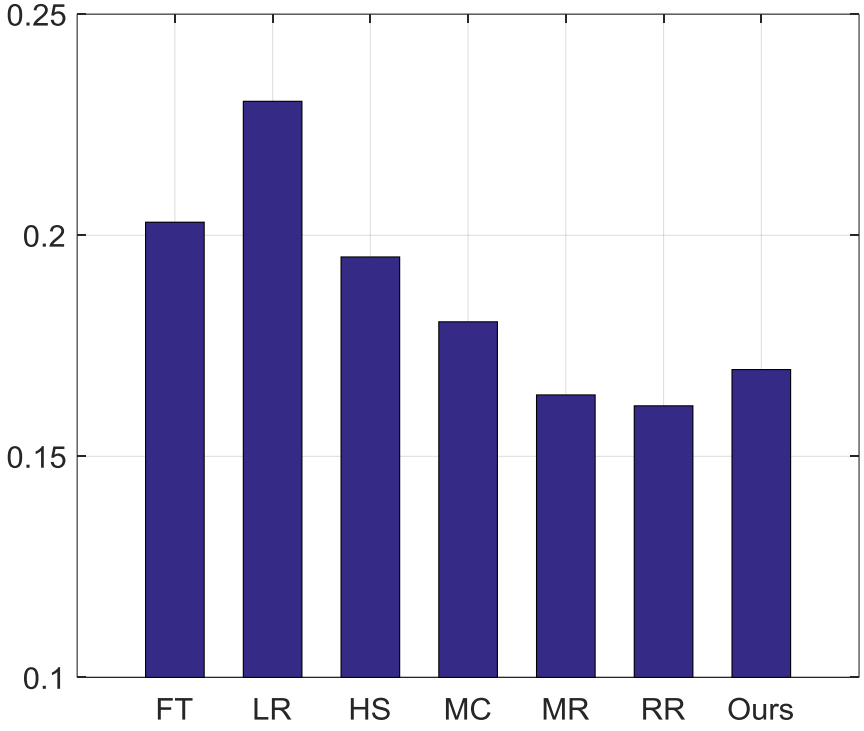

Figure 5.16 MAE scores on the SED2 dataset.

We then conduct the qualitative evaluation of our proposed BSFE method, and the visual saliency map examples are shown in Figure 5.17. It depicts that BSFE achieves the best qualitative performance against the comparison methods. For example, Figure 5.17(b), (c), (d), (g), (h) and (j) involve images with low contrast salient objects, in which BSFE successfully extracts the whole salient object, while all of the comparison methods miss part of the object more or less. Figure 5.17(a), (e), (f), (i), (k) and (l) involve images with complex foreground / background patterns, in which BSFE managed to recognize the salient object from the complex background (even for images with two objects such as Figure 5.17(k) and (l)), while most of the comparison methods fail to correctly detect the salient object.

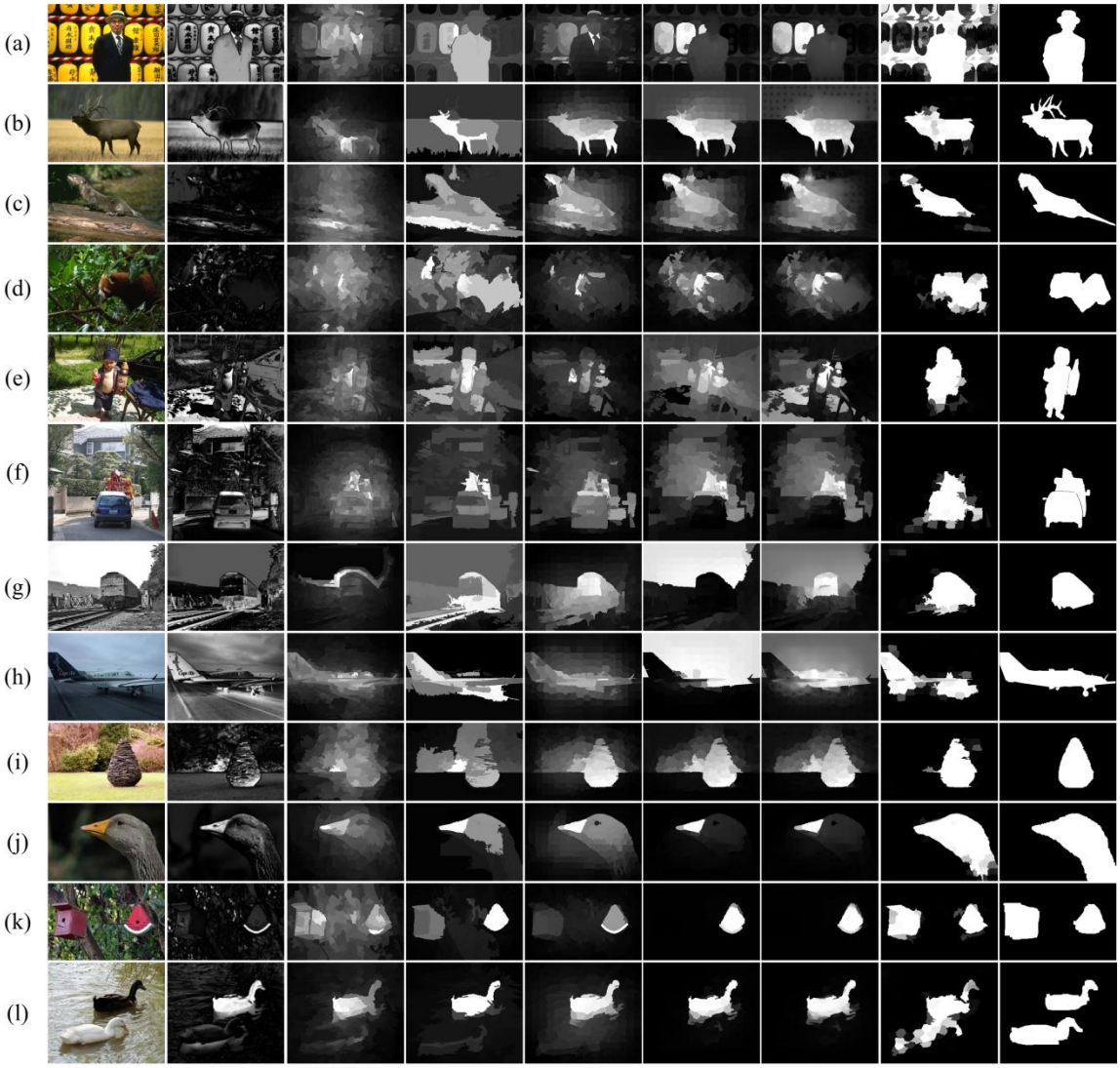

Figure 5.17 Saliency map examples of state-of-the-art methods against our BSFE method. (a) – (l): Image case IDs.

5.6 Saliency Detection with Multi-Dimensional Features Using DNN Based Dense and Sparse Labeling (DSL)

In this section, the DSL method will be introduced in detail. As mentioned in section 5.2, our DSL method has three major steps, namely DL, SL and DC. The complete flowchart of DSL is shown in Figure 5.18. Considering the topological structure of the three steps, two independent training datasets T_1 and T_2 are used, in which T_1 is used for DL and SL, and T_2 is used for DC.

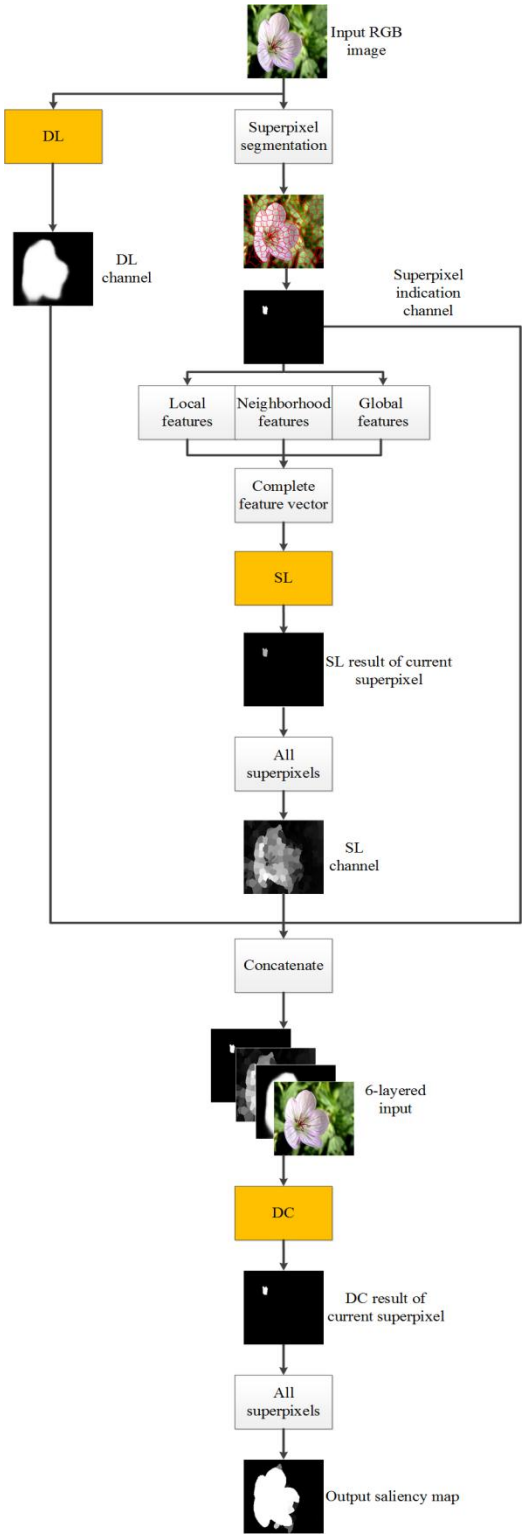

Figure 5.18 Flowchart of our DSL method. The three major steps DL, SL and DC are highlighted in yellow. An input image is first processed by DL and SL, respectively; the resulting initial saliency estimations are then concatenated with the image RGB channels and the superpixel indication channel to form the 6-channel input of DC, which is used to generate the final saliency map.

5.6.1 Dense Labeling for Initial Saliency Estimation

Dense labeling is a category of classification in which each pixel in the input image is assigned a label that indicates the type of object it most likely belongs to. Saliency detection can be treated as a binary dense labeling problem, since the salient (foreground) and background regions can be seen as two separate objects.

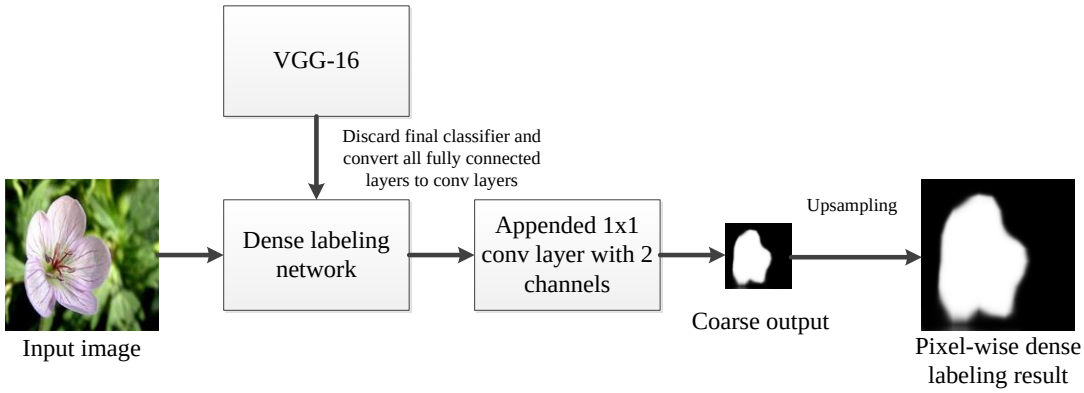

Figure 5.19 Flowchart of the DL step.

The flowchart of our DL network is shown in Figure 5.19. It is inspired by [150], which has achieved state-of-the-art performance in dense labeling tasks such as semantic segmentation. The network architecture is shown in Table 5.3. The main differences between DL and a normal CNN are that DL takes enlarged input images (up to 384*384), and the last few originally fully-connected (fc) layers are converted to 1*1 convolutional layers. As a result, the heatmaps (instead of scalar labels) of foreground and background can be directly generated at layer conv8, both with size 12*12. We then apply the bilinear interpolation to upsample the heatmaps from 12*12 (M_{conv8}) to 224*224 ($M_{deconv32}$), which is the input size of the following DC step. For each to-be-interpolated pixel on $M_{deconv32}$, its upsampled value is calculated by bilinear interpolation of its closest four values on M_{conv8} , as indicated in Figure 5.20:

$$M^{l}_{deconv32}(x,y) = \frac{1}{(x_2-x_1)(y_2-y_1)}\begin{pmatrix} M^{l}_{conv8}(x_1,y_1)(x_2-x)(y_2-y) \\ +M^{l}_{conv8}(x_2,y_1)(x-x_1)(y_2-y) \\ +M^{l}_{conv8}(x_1,y_2)(x_2-x)(y-y_1) \\ +M^{l}_{conv8}(x_2,y_2)(x-x_1)(y-y_1) \end{pmatrix} \tag{5.8}$$

where $l \in [0,1]$ stands for the salient (foreground) layer and background layer. Note that all coordinates are normalized to [0,1] to facilitate calculation. After that, similar to the softmax regression in normal CNNs, we take each two pixels on $M_{deconv32}$ with the same x and y coordinates (but at different layers) as a pair, and apply the softmax function on them:

$$M^{l}_{sm}(x,y) = \frac{\exp\left(M^{l}_{deconv32}(x,y)\right)}{\sum_{k=0}^{1}\exp\left(M^{k}_{deconv32}(x,y)\right)}. \tag{5.9}$$

The L2 loss is then computed between the pixel-wise ground truth G and M_{sm}:

$$J_{DL} = \sum_{l=0}^{1}\sum_{x=1}^{X}\sum_{y=1}^{Y}\left(G(x,y)==l\right)\log\left(M^{l}_{sm}(x,y)\right), \tag{5.10}$$

where "==" means the logical "equal to". Equation (5.10) is later used in the back-propagation for training.

Table 5.3 Architecture of our DL network

Layer	Type	Output Size	Conv (size, channel, pad)	Max Pooling
input	in	384*384*3	N/A	N/A
conv1_1	c+r	384*384*64	3*3,64,1	N/A
conv1_2	c+r+p	192*192*64	3*3,64,1	2*2
conv2_1	c+r	192*192*128	3*3,128,1	N/A
conv2_2	c+r+p	96*96*128	3*3,128,1	2*2
conv3_1	c+r	96*96*256	3*3,256,1	N/A
conv3_2	c+r	96*96*256	3*3,256,1	N/A
conv3_3	c+r+p	48*48*256	3*3,256,1	2*2
conv4_1	c+r	48*48*512	3*3,512,1	N/A
conv4_2	c+r	48*48*512	3*3,512,1	N/A

conv4_3	c+r+p	24*24*512	3*3,512,1	2*2
conv5_1	c+r	24*24*512	3*3,512,1	N/A
conv5_2	c+r	24*24*512	3*3,512,1	N/A
conv5_3	c+r+p	12*12*512	3*3,512,1	2*2
conv6	c+r+d	12*12*4096	7*7,4096,3	N/A
conv7	c+r+d	12*12*4096	1*1,4096,0	N/A
conv8	c	12*12*2	1*1,2,0	N/A
deconv32	us	384*384*2	N/A	N/A
loss	sm+log	1*1	N/A	N/A

Annotations - in: input layer; c: convolutional layer; r: ReLU layer; p: pooling layer; d: dropout layer; us: upsampling layer; sm: softmax layer; log: log loss layer.

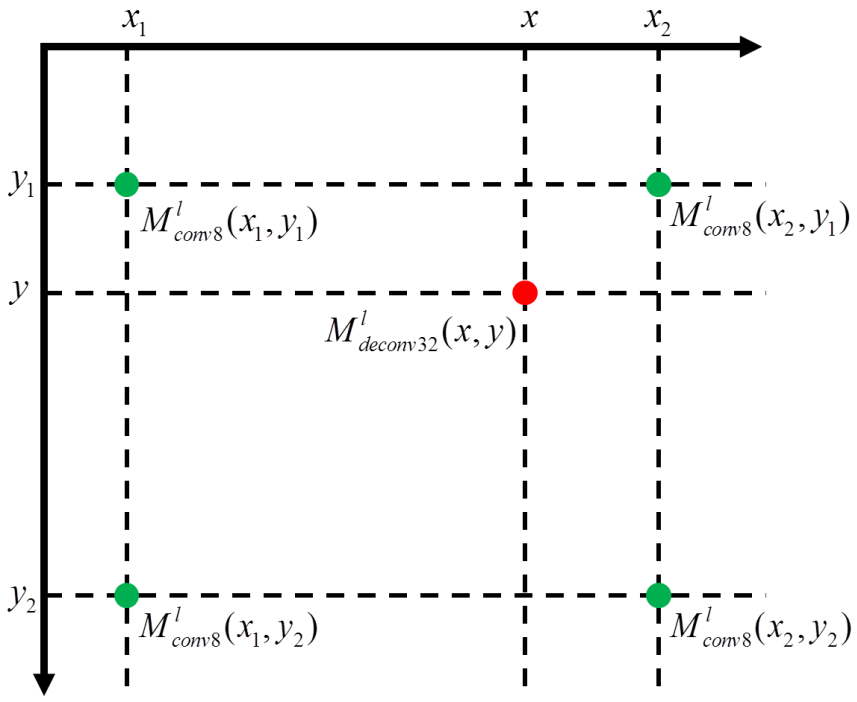

Figure 5.20 Bilinear interpolation from the conv8 layer to the deconv32 layer.

As mentioned at the beginning of section 5.6, the DL network is trained by the training set T_1. After desired validation results are obtained, it is used to test the training set T_2, the results of which are then used as part of the 6-channeled inputs in training the DC step, as Figure 5.18 shows. Figure 5.21 illustrates example outputs of DL. It is observed that DL is capable of producing accurate contours of the salient object, which contains much more boundary information than the bounding box approximation in

[147]. In addition, it also has shown high robustness in various challenging scenarios, such as low contrast images (Figure 5.21(c)) and complex images (Figure 5.21(d)).

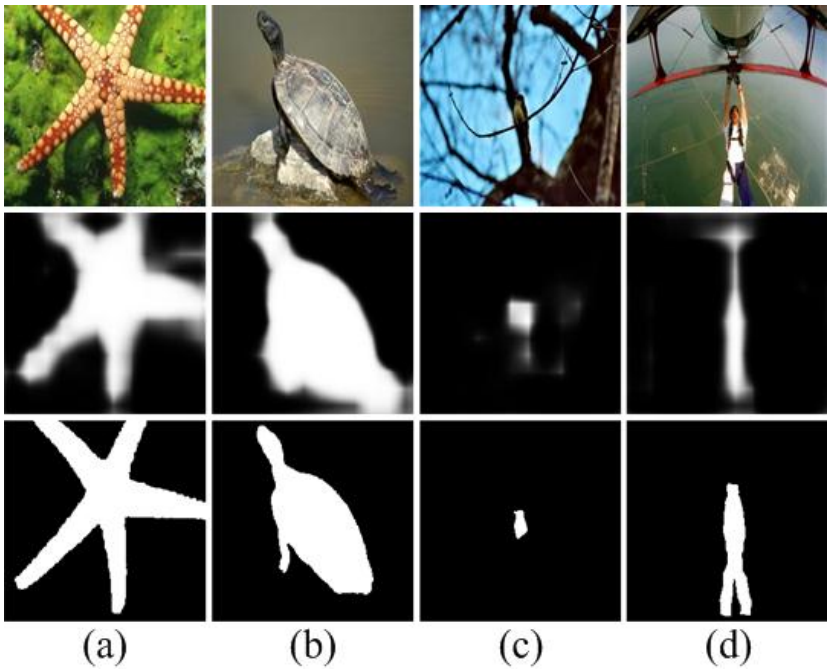

Figure 5.21 Example outputs of the DL step. First row: input images; second row: outputs of the DL network; third row: ground truth.

5.6.2 Sparse Labeling for Initial Saliency Estimation

Similar to the DL step which produces initial saliency estimation with macro object contours, the SL step produces initial saliency estimation with low-level image features.

The idea of the SL step is to conduct superpixel-wise sparse labeling of the image based on its corresponding low-level features. Each image is first segmented into superpixels by the SLIC method [166]. We adopt a zoom-out-like feature fusion of each superpixel [149], which involves 708 local features, 204 neighborhood features, and 4096 global features (5,008 features in total for each superpixel). The three different types of features are introduced below.

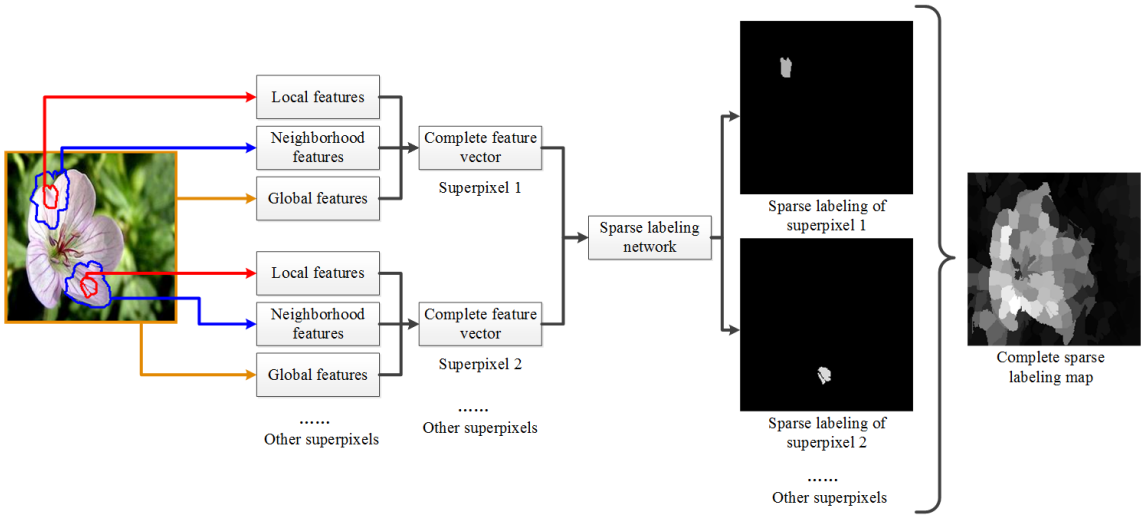

Figure 5.22 Flowchart of the SL step. The input image after superpixel segmentation is processed by local, neighborhood and global feature extractions for the complete feature vector. The sparse labeling network then intakes the complete feature vector and conducts image-feature-based initial saliency estimation.

(1) Local Features

The local features are on the smallest scope in our feature extraction, which focus on the current superpixel itself, as the red regions in Figure 5.22 indicate. Due to the narrow scope, the local features tend to have large variance among neighboring superpixels. There are 708 local features in total that we have adopted, including 204 color features, 4 location features, and 500 local CNN features.

Color: We first extract the bounding box of the current superpixel, and then calculate its histograms for each of the three channels in both RGB and L*a*b color spaces, with 32 color bins each. In addition, the mean and variance for each of the three channels in the two color spaces are also calculated. This yields 32*3*2 + 2*3*2 = 204 color features.

Location: We compute the min / max x and y coordinates of the current superpixel's bounding box, and conduct normalization to the size of the image. This yields 4 location features in the range of [0, 1].

Local CNN: The last part of local feature is a representation of the current superpixel by a local CNN, which is fine-tuned from the LeNet model for hand-written digit recognition [183]. Table 5.4 shows the architecture of the local CNN, which has three convolutional layers separated by batch normalization [184], max pooling and ReLU layers. It takes the bounding box of the current superpixel in the L*a*b color space as input (resized to 28*28*3), and outputs a binary label that indicates the current superpixel being salient or background. We select the output of conv3, which is the activation value of the last fully connected layer fc4, as the local CNN feature. This yields the 500 CNN features.

Table 5.4 Architecture of our local CNN

Layer	Type	Output Size	Conv (size, channel, pad)	Max Pooling
input	in	28*28*3	N/A	N/A
conv1	c+b+p	12*12*20	5*5,20,0	2*2
conv2	c+b+p	4*4*50	5*5,50,0	2*2
conv3	c+b+r	1*1*500	4*4,500,0	N/A
fc4	fc+r	1*1*2	1*1,2,0	N/A
loss	sm+log	1*1	N/A	N/A

Annotations - in: input layer; c: convolutional layer; b: batch normalization layer; p: pooling layer; r: ReLU layer; fc: fully connected layer; sm: softmax layer; log: log loss layer.

(2) Neighborhood Features

The neighborhood features are on the second scope in our feature extraction, which focuses on the neighboring regions of the current superpixel. The neighboring region is defined as the second order neighboring superpixels of the current superpixel, as the blue regions in Figure 5.22 indicate. They are designed to reflect an intermediate level of features of the current superpixel, which are more enriched than the local features, but are less macro-scoped than the global features. Due to its definition, the neighborhood features are expected to have lower variance among different superpixels than the local

features. We adopt the same set of color features defined in the previous section as the neighborhood features, which yields 204 features.

(3) Global Features

The global features consist of representations of the whole image, as the yellow region (outer boundary) in Figure 5.22 indicates. We use a CNN designed for ImageNet classification to generate the global features. By considering the overall performance, the VGG-16 model [161] is adopted, which is the same model used in our DC step (see section 5.6.3 for detailed discussion). Images are resized to 224*224 before being fed into the network, and the 1*1*4,096 activation value of the last fully connected layer is treated as the global feature. Following [149], we directly use the pre-trained network without fine-tuning.

(4) SL Network Training

By feature extraction and concatenation of the three steps above, a 1*5,008 feature vector will be generated per superpixel per image. We then establish the SL network with three fully connected layers (see section 5.7.5 for detailed discussion), which takes the feature vectors as inputs, and output a scalar label indicating the saliency of the current superpixel. After training for enough epochs, the SL network is used to generate the low-level feature based initial saliency channel for the next DC step.

5.6.3 Sparse Labeling for Final Saliency Map

While the DL and SL steps are designed to provide coarse initial saliency estimations, the DC step is designed to generate the final saliency map with superpixel-wise binary sparse labeling, i.e. obtain the saliency of each individual superpixel in the image via DNN-based classification, and then integrate them together to form the complete final

saliency map, as shown in Figure 5.18. Considering the overall performance, we adopt the VGG-16 [161] as the baseline model of our DC network (see section 5.7.6 for detailed discussion). Table 5.5 shows the architecture of the DC network. The input structure of DC, being one of our key novelties, is 6-channeled data with fixed size as 224*224*6. The first three channels are the RGB data from the image; the fourth and fifth channels are the initial saliency estimations from the DL and SL steps, respectively (both resized to 224*224); and the sixth channel is the superpixel indication channel, which precisely marks the current to-be-classified superpixel, as the "Superpixel indication channel" in Figure 5.18 indicates.

Table 5.5 Architecture of our DC network

Layer	Type	Output Size	Conv (size, channel, pad)	Max Pooling
input	in	224*224*6	N/A	N/A
conv1_1	c+b+r	224*224*64	3*3,64,1	N/A
conv1_2	c+b+r	112*112*64	3*3,64,1	2*2
conv2_1	c+b+r	112*112*128	3*3,128,1	N/A
conv2_2	c+b+r	56*56*128	3*3,128,1	2*2
conv3_1	c+b+r	56*56*256	3*3,256,1	N/A
conv3_2	c+b+r	56*56*256	3*3,256,1	N/A
conv3_3	c+b+r	28*28*256	3*3,256,1	2*2
conv4_1	c+b+r	28*28*512	3*3,512,1	2*2
conv4_2	c+b+r	28*28*512	3*3,512,1	N/A
conv4_3	c+b+r	14*14*512	3*3,512,1	2*2
conv5_1	c+b+r	14*14*512	3*3,512,1	N/A
conv5_2	c+b+r	14*14*512	3*3,512,1	N/A
conv5_3	c+b+r	7*7*512	3*3,512,1	2*2
fc6	fc+r	1*1*4096	7*7,4096,0	N/A
fc7	fc+r	1*1*4096	1*1,4096,0	N/A
fc8	fc+r	1*1*2	1*1,2,0	N/A
loss	sm+log	1*1	N/A	N/A

Annotations - in: input layer; c: convolutional layer; b: batch normalization layer; p: pooling layer; r: ReLU layer; fc: fully connected layer; sm: softmax layer; log: log loss layer.

To obtain the superpixel indication channel, we first segment the image into superpixels, also by the SLIC method used in section 5.6.2. The to-be-classified superpixel is then selected and marked on a 224*224 black background, i.e. assigning

the pixels within the superpixel as maximum intensity, while all the other pixels remain zero. Note that the superpxiel indication channel is the only channel to differentiate the inputs of different superpixels from the same image. Hence, provided that the number of images and number of superpixels per image are N_{im} and N_{sp}, respectively, there will be $N_{im} \cdot N_{sp}$ samples in total.

Let Y_i be the activation value of the fc8 layer for the *i*-th superpixel, whose size is changed from the originally 1000 to 2, indicating binary classification (salient or background). A softmax loss layer is applied afterwards to compute the logarithm loss, with N_{sp} as the batch size:

$$J_{DC} = -\frac{1}{N_{sp}} \sum_{i=1}^{N_{sp}} \left[G_i \log P_i + (1 - G_i) \log(1 - P_i) \right] + \lambda_C \sum_j \left(W_j^T W_j \right), \tag{5.11}$$

where

$$P_i = \frac{\exp\left(Y_i(1)\right)}{\exp\left(Y_i(0)\right) + \exp\left(Y_i(1)\right)} \tag{5.12}$$

is the softmax probability of i being salient; $G_i \in [0,1]$ is the ground truth label of i; λ_C is the weight decay parameter; j stands for the layers with trainable weights of the DC network; and W_j is the weight vector of layer j.

We then train DC by the T_2 dataset, as mentioned at the start of section 5.6, with N_{sp} samples per batch and N_{im} batches in total. As for testing, the probability P_i in (5.12) is adopted as the saliency value for the superpixel *i*, which is assigned to all the pixels within *i*. And the final saliency map is formed when all of the superpixels in the

current image have obtained their corresponding saliency values, as indicated in Figure 5.18.

The major advantage of DC is attributed to its 6-channeled input structure. Unlike existing DNN-based methods like [90], [91] that only use RGB or other features from the current image itself, DC integrates two coarse guiding channels via dense labeling (DL) and sparse labeling (SL). The two guiding channels provide reliable prior knowledge with learned high-level features from the entire training dataset, and can accurately approximate the salient region as well as exclude false salient proposals. The 6-channeled input structure also contains the superpixel indication channel, which directly and precisely marks the current to-be-classified superpixel, unlike [91] which only vaguely indicates the superpixel by putting it to the image center. The examples in Figure 5.23 exhibit the combined strength of the DL, SL and DC steps. Note that DL and SL contribute complementarily to the DC step (i.e. the final output of DSL), especially in cases where one of DL or SL encounters difficulty in estimating the initial saliency accurately, as seen in Figure 5.23(c) and Figure 5.23(d). The combination of DL and SL thus significantly increases the overall robustness of DSL.

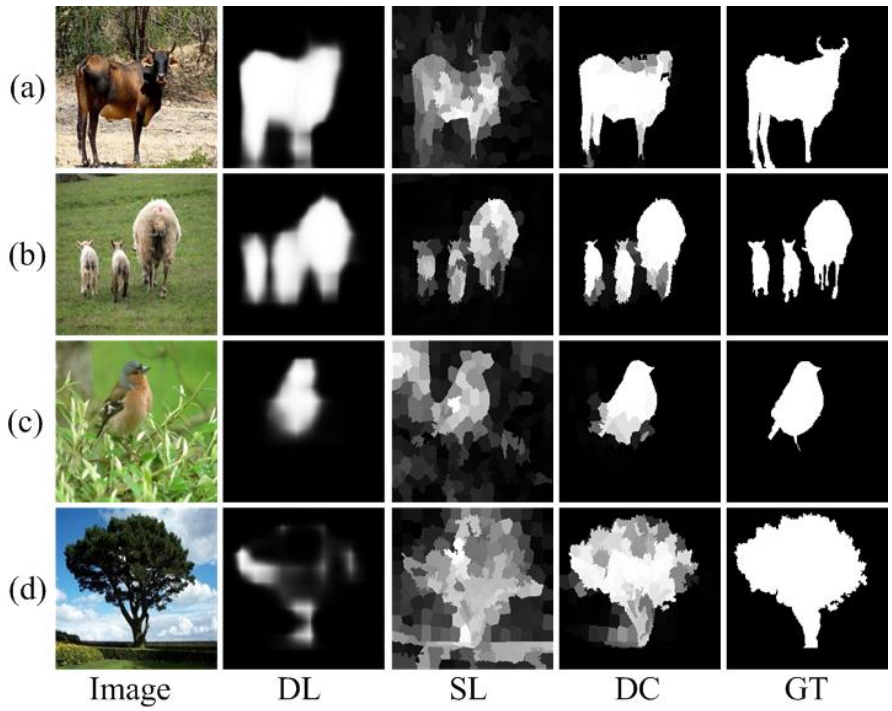

Figure 5.23 Example outputs of the DL, SL, and DC steps. Note that DL and SL contributes complementarily to the DC step, which generates the final output of the proposed DSL method.

5.7 Experimental Results of DSL

In this section, we present the experimental results of our proposed DSL method. We first introduce the datasets, evaluation metrics and implementation details that we used, and then systematically analyze the parameters for each of three steps in our method, namely DL, SL and DC. We then compare the contributions of the three steps in our proposed DSL method. After that, we present the comparison experiments against sixteen state-of-the-art saliency detection methods, with ten conventional methods and six learning based methods. Finally, we present the efficiency and limitation of our DSL method.

5.7.1 Datasets

As mentioned at the beginning of section 5.6, since DL and SL are both serially connected to DC (Figure 5.18), it is necessary to use two independent training sets for DL / SL and DC respectively, in order to conduct fair trainings.

For the training of DL and SL, we use the DUT-OMRON dataset [52], which contains 5,168 manually selected high quality images and corresponding pixel-wise ground truth. We randomly select 80% of the images for training, and the rest 20% images for validation.

For the training of DC, we use the MSRA10K dataset [63], which contains 10,000 randomly chosen images from the MSRA dataset [10], and their corresponding pixel-wise ground truth. To make the comparison with state-of-the-art methods fair, we follow [91] and randomly choose 80% of the images for training, and the rest 20% images for validation.

For testing, we adopt six well-recognized public datasets, namely ECSSD [69], PASCAL-S [172], SED1 [171], SED2 [171], THUR15K [185], and HKU-IS [92]. The ECSSD dataset contains 1,000 complex images with diversified contexts. The PASCAL-S dataset is a subset of the PASCAL-S VOC segmentation challenge [173], which contains 850 images with highly challenging backgrounds. The SED1 and SED2 are two datasets designed for saliency detection, with 100 images each; the images of SED1 contain one salient object, while the images of SED2 contain two salient objects. The THUR15K dataset contains 15,000 images, among which we only use the 6,233 images with pixel-wise ground truth. For the HKU-IS dataset, we only use the 1,447 images in

the test set that have no overlap with any of our comparison methods' training set in our following experiments.

5.7.2 Evaluation Metrics

Following a recent saliency detection benchmark [8], we choose the precision-recall (PR) curve, F-measure, and mean absolute error (MAE) as our evaluation metrics.

The precision and recall values are obtained by binarizing the saliency map with integer thresholds between 0 and 255. The precision value equals to the ratio of retrieved salient pixels to all the pixels retrieved, while the recall value equals to the ratio of retrieved salient pixels to all salient pixels in the image. The PR curve is plotted by the precision and recall values at each threshold point.

The F-measure is a weighted average between precision and recall, which is calculated as:

$$F_\beta = \frac{(1+\beta^2)\, precision \cdot recall}{\beta^2\, precision + recall}, \tag{5.13}$$

where β^2 is set to 0.3 based on most existing methods. As suggested in [174], the average F-measure of a PR curve equals to its maximum single-point F-measure.

The MAE is the mean of the absolute difference between the saliency map S and the pixel-wise ground truth G:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|S(i) - G(i)\right|. \tag{5.14}$$

Different to precision, recall and F-measure, smaller MAE means higher performance.

5.7.3 Implementation

Our method is implemented on MatConvNet [186], which is a MATLAB toolbox of CNN with various extensibilities. The machine used for our experiments is a PC with Intel 6-Core i7-5820K 3.3GHz CPU, 64GB RAM, GeForce GTX TITAN X 12GB GPU, and 64-bit Ubuntu 14.04.3 LTS. Software dependencies include CUDA 7.0 and cuDNN v3. All images are stored on SSD, which accelerates reading speed. The source code of our proposed DSL method is available online: https://github.com/yuanyc06/dsl.

5.7.4 Parameter Analysis of the DL Step

The DL network is trained on the DUT-OMRON dataset for 50 epochs, with 50-point logarithm space between 10^{-3} and 10^{-4} as the learning rate. As described in section 5.6.1, the images are resized to 384*384*3 before fed into the network.

To evaluate the network architecture of DL, we compare it against two state-of-the-art dense labeling models extended from [150], namely FCN-8s and FCN-16s. We fine-tune our DL network on each of the three models, and record the performance of the three architectures on the validation set of the 50^{th} epoch. The results are shown in Table 5.6.

It is apparent that the proposed DL architecture has the optimal performance against the other two models, largely due to its less likelihood of over-fitting. Since the original object detection task in [150] was performed on a relatively large dataset (~30K images on the VOC2011 dataset), it was reasonable that the more complex models had higher performances (i.e. FCN-32s < FCN-16s < FCN-8s). On the other hand, in our DL step the training dataset is relatively small (only 5,168 images), thus more complex models

are more vulnerable to over-fitting. As a result, it is the less complex model DL (FCN-32s) that performs the best.

Table 5.6 Performances of the DL network against two state-of-the-art dense labeling models

Model	F-Measure	MAE
FCN-8s	0.670	0.149
FCN-16s	0.727	0.137
DL	**0.747**	**0.128**

*The F-measures and MAEs are recorded on the validation set at the 50th training epoch. The best results are marked in **red**.*

5.7.5 Parameter Analysis of the SL Step

There are two networks to train for the SL step, namely the local CNN and the SL network itself. We randomly select 2,000 images from the DUT-OMRON dataset for the local CNN, and the rest 3,168 images for the SL network. Both of the networks use 80% of their assigned images for training, and the rest 20% for validation. They are both trained for 50 epochs, with 50-point logarithm space between 10^{-2} and 10^{-4} as the learning rate. We use the SLIC [166] method to generate the superpixels required, with 200 superpixels per image. As described in section 5.6.2, the input of the local CNN are superpixel patches resized to 28*28*3, while the input of the SL network are 1*5,008 feature vectors of the superpixels.

The local CNN is fine-tuned from LeNet [183], and the SL network is trained from scratch (since no baseline model available). To determine the optimal network architecture for SL, we change the network layer number (#layer) and parameter number per layer (#param) 2-dimensionally, and record the validation performances on the 50th training epoch, as shown in Table 5.7. The configuration that gives the best performance is #layer = 3 and #param = 2048, which are adopted in our following experiments.

After determining the network architecture of SL, we further analyze the influence of its three types of features (i.e. local, neighborhood and global features) to the overall performance of our DSL method. The analysis is conducted on the two challenging datasets ECSSD and PASCAL-S, and we use seven different combinations of the features to train the SL network (the feature vector of SL is changed accordingly), and use the corresponding feature combinations in the testing processes. Table 5.8 shows the evaluation results, in which using all three types of features contributes to the best performance in terms of both F-measure and MAE on both of the datasets. We thus adopt all three types of features for the SL step.

Table 5.7 Performances of the SL network with different layer number (#layer) and parameters per layer (#param)

Configuration	**F-Measure**	**MAE**
#layer=3, #param=1024	0.664	0.182
#layer=3, #param=2048	**0.670**	**0.171**
#layer=3, #param=4096	0.666	0.178
#layer=4, #param=1024	0.661	0.180
#layer=4, #param=2048	0.654	0.186
#layer=4, #param=4096	0.652	0.193

*The F-measures and MAEs are recorded on the validation set at the 50th training epoch. The best results are marked in **red**.*

Table 5.8 Performances of DSL with different SL feature combinations

Dataset	Feature of SL	F-Measure	MAE
ECSSD	local	0.783	0.213
	neighborhood	0.778	0.224
	global	0.795	0.181
	local + neighborhood	0.789	0.174
	neighborhood + global	0.801	0.166
	local + global	0.804	0.158
	all	**0.808**	**0.126**
PASCAL-S	local	0.777	0.178
	neighborhood	0.770	0.195
	global	0.782	0.143
	local + neighborhood	0.780	0.162
	neighborhood + global	0.786	0.136
	local + global	0.788	0.131
	all	**0.791**	**0.122**

*The best results are marked in **red**.*

5.7.6 Parameter Analysis of the DC Step

The DC network is trained on the MSRA10K dataset. We first feedforward MSRA10K through DL and SL to obtain the two initial saliency channels of its input images, and then form the 6-channeled inputs for DC. The DC network is trained for 20 epochs, with 20-point logarithm space between 10^{-2} and 10^{-4} as the learning rate. The superpixels are generated by the SLIC method as well, with 200 superpixels per image.

To determine the best baseline model, we fine-tune the DC network on three state-of-the-art image classification models, namely AlexNet [144], VGG-16 [161], and GoogLeNet [146]. We record their performances on the two challenging datasets ECSSD and PASCAL-S in Table 5.9. It is observed that VGG-16 has the best overall performance than the other two models, and previous works have proved its steadiness and robustness in various computer vision tasks [93], [150], [187], [188]. We thus adopt VGG-16 as our baseline model for the DC step.

Table 5.9 Performances of the DC step with different baseline models on the two challenging datasets ECSSD and PASCAL-S

Dataset	Model	F-Measure	MAE
ECSSD	AlexNet	0.802	0.133
	VGG-16	**0.808**	**0.126**
	GoogLeNet	0.807	0.129
PASCAL-S	AlexNet	0.782	0.128
	VGG-16	**0.791**	**0.122**
	GoogLeNet	0.789	0.127

*The best results are marked in **red**.*

5.7.7 Contribution Comparison

Next, we examine the contributions of the three steps (i.e. DL, SL and DC) in improving the performance of our method. We take the "pad-and-center" method in [91] as the comparison baseline, and compare five different configurations below:

(1) Baseline: the local pad-and-center model in [91]; the network takes padded image as input (224*224*3) (without the superpixel indication channel);

(2) DC only: the input of DC is thus 224*224*4 (with the superpixel indication channel, but without the DL and SL channels);

(3) DL and DC: the input of DC is thus 224*224*5 (with the superpixel indication channel, but without the SL channel);

(4) SL and DC: the input of DC is thus 224*224*5 (with the superpixel indication channel, but without the DL channel);

(5) Complete DSL model: the DC network takes the 224*224*6 input with all of the 6 channels.

Similarly to the previous section, we record the performances of the five configurations above on the two challenging datasets ECSSD and PASCAL-S. The results are listed in Table 5.10. We see that the complete DSL framework (Configuration v: DL+SL+DC) notably outperforms the other four configurations, which indicates that DL, SL and DC all have significant contributions in improving the overall performance of DSL.

Table 5.10 Performances of different design option configurations on the two challenging datasets ECSSD and PASCAL-S

Dataset	Configuration	F-Measure	MAE
ECSSD	Config i: Baseline	0.724	0.187
	Config ii: DC only	0.750	0.171
	Config iii: DL+DC	0.788	0.147
	Config iv: SL+DC	0.772	0.162
	Config v: DL+SL+DC	**0.808**	**0.126**
PASCAL-S	Config i: Baseline	0.681	0.168
	Config ii: DC only	0.729	0.148
	Config iii: DL+DC	0.777	0.140
	Config iv: SL+DC	0.759	0.143
	Config v: DL+SL+DC	**0.791**	**0.122**

*The best results are marked in **red**.*

5.7.8 Evaluation Against Conventional Methods

Next, we compare our proposed DSL method with ten state-of-the-art conventional saliency detection methods (no learning process), namely SF [70], GR [83], MC [54], MR [52], DSR [80], HS [69], RBD [53], RR [1], BSCA [72], and BL [87]. All of the ten methods are published after 2012, and the last three methods are recently published in 2015. As mentioned in section 5.7.1, the experiments are conducted on the six datasets ECSSD, PASCAL-S, SED1, SED2, THUR15K and HKU-IS. The precision-recall curves are shown in Figure 5.24 to Figure 5.29, and the quantitative evaluation results are shown in Table 5.11.

The first thing we notice is that DSL not only achieves the best performance on all of the dataset in terms of both F-measure and MAE, but also exceeds the comparison methods with dominant advantages. We first analyze the two challenging datasets ECSSD and PASCAL-S, where DSL's PR curves are greatly higher than the comparison methods, and its F-measures and MAEs have shown significantly large gaps against the second best methods. To be specific, its F-measures are 12.5% and 18.2% higher than the second best (0.808 to 0.718, and 0.791 to 0.669), and its MAEs are 78.6% and 65.6% lower than the second best (0.126 to 0.225, and 0.122 to 0.202). We attribute the greatly improved performance of DSL to its integrated structure of multiple DNNs, in which both dense and sparse labeling show their strength in extracting the high-level features of the image, as well as their combined advantage that further boost the saliency classification accuracy.

DSL behaves similarly on the other four datasets, where it shows dominant advantages on both PR curves and evaluation metrics against all of the comparison

methods. It is mentionable that the advantage of DSL on SED2 is not as significant as those on the other datasets. This is mainly due to the single-object training set we used, while all of the images in SED2 contain double salient objects.

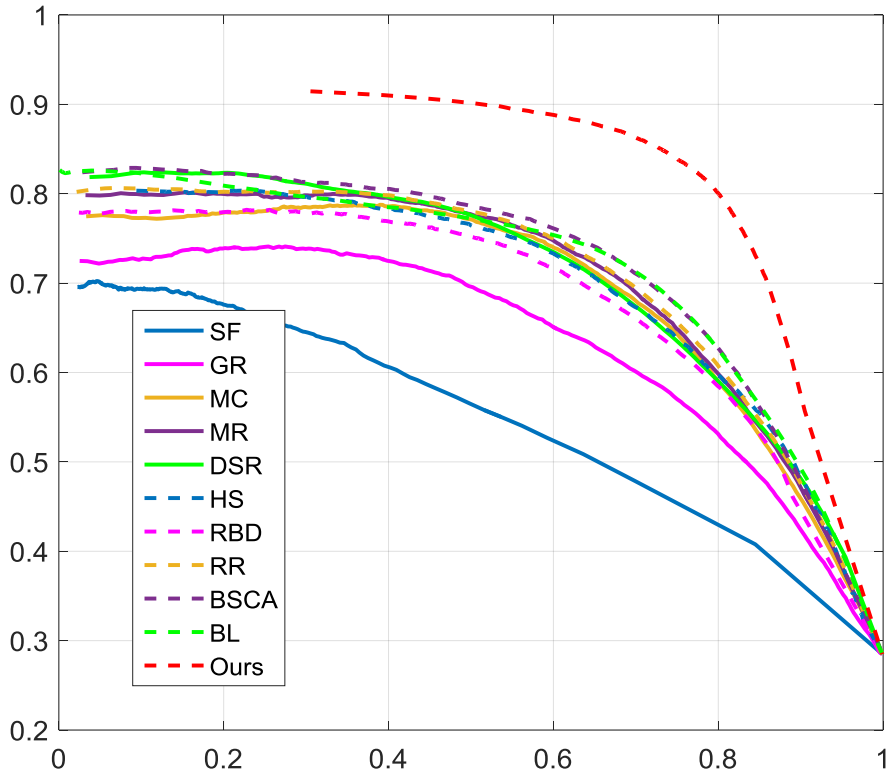

Figure 5.24 Precision-recall curves against conventional methods on the ECSSD dataset.

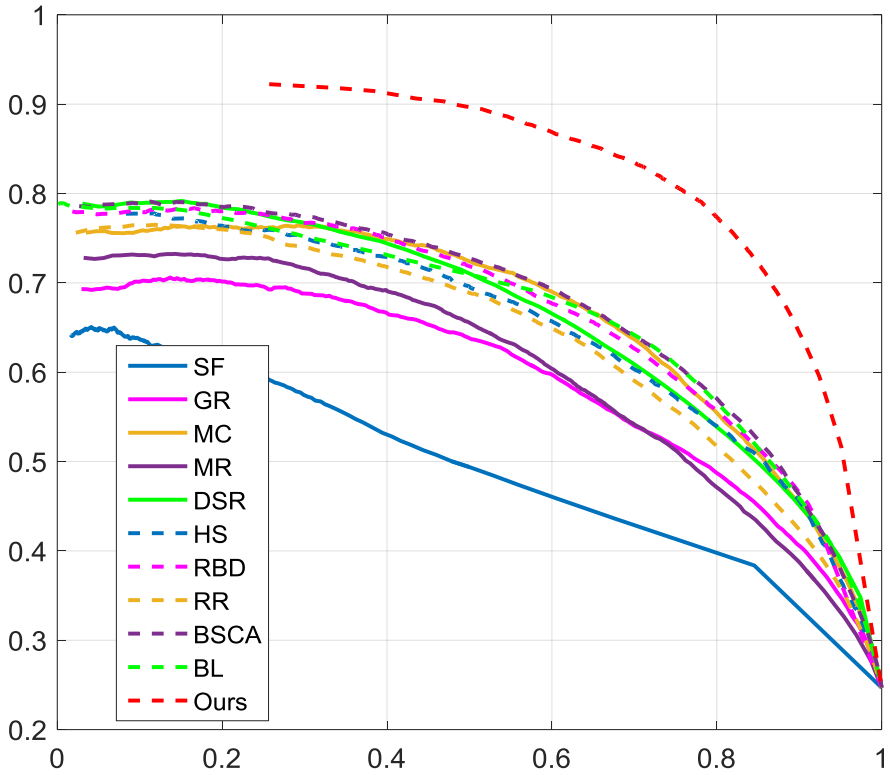

Figure 5.25 Precision-recall curves against conventional methods on the PASCAL-S dataset.

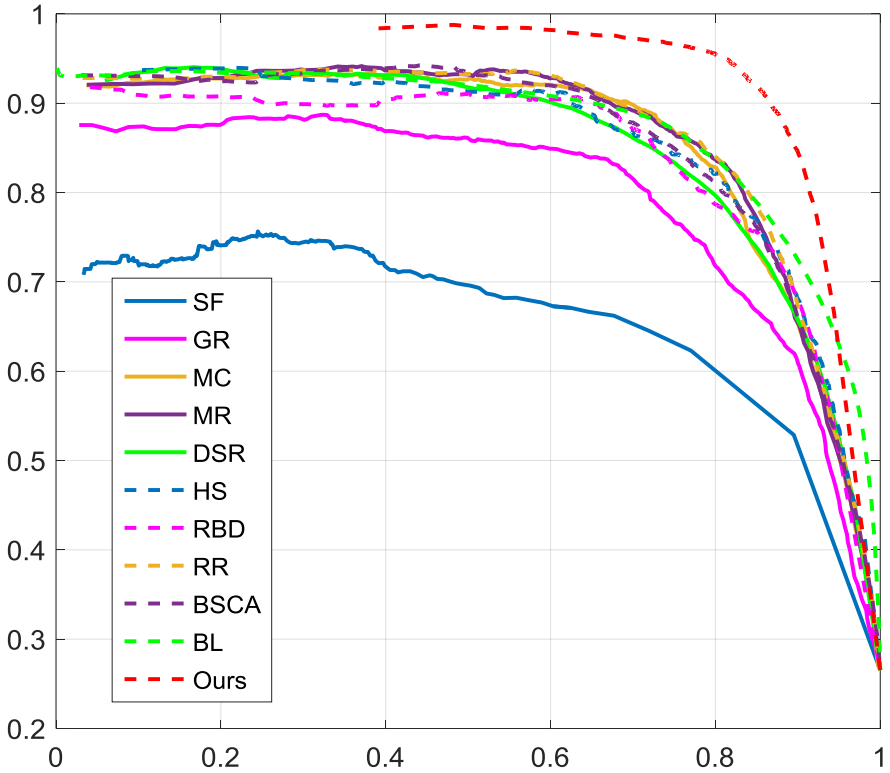

Figure 5.26 Precision-recall curves against conventional methods on the SED1 dataset.

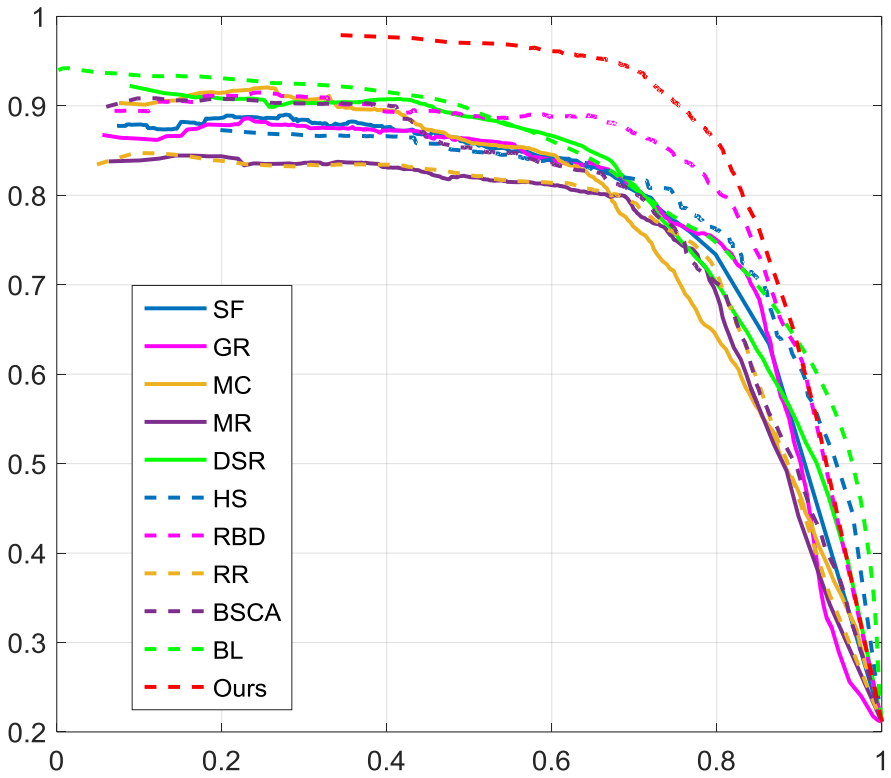

Figure 5.27 Precision-recall curves against conventional methods on the SED2 dataset.

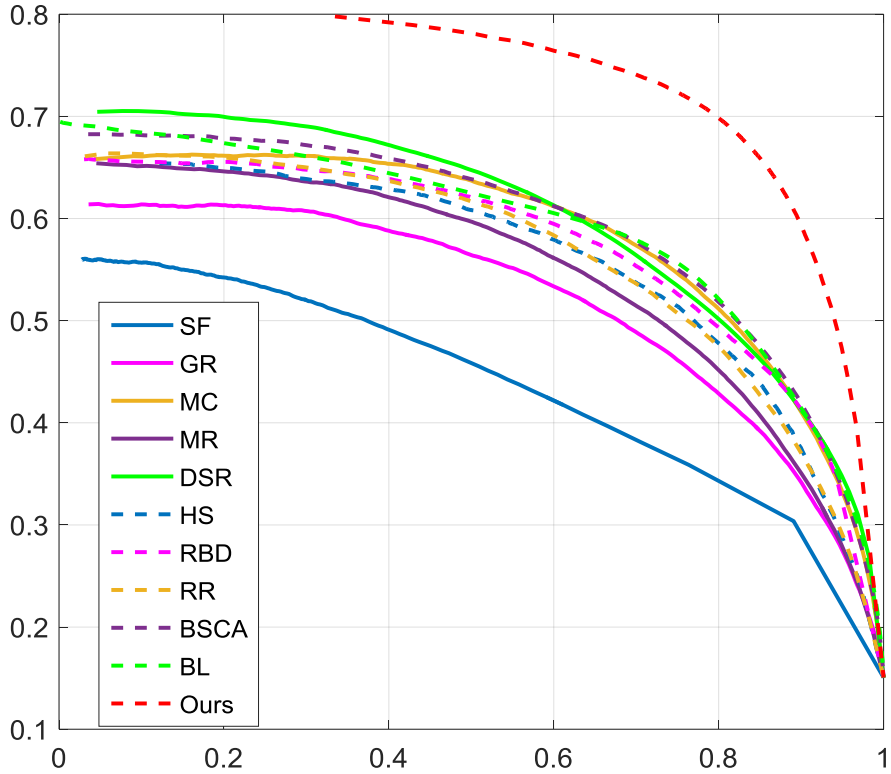

Figure 5.28 Precision-recall curves against conventional methods on the THUR15K dataset.

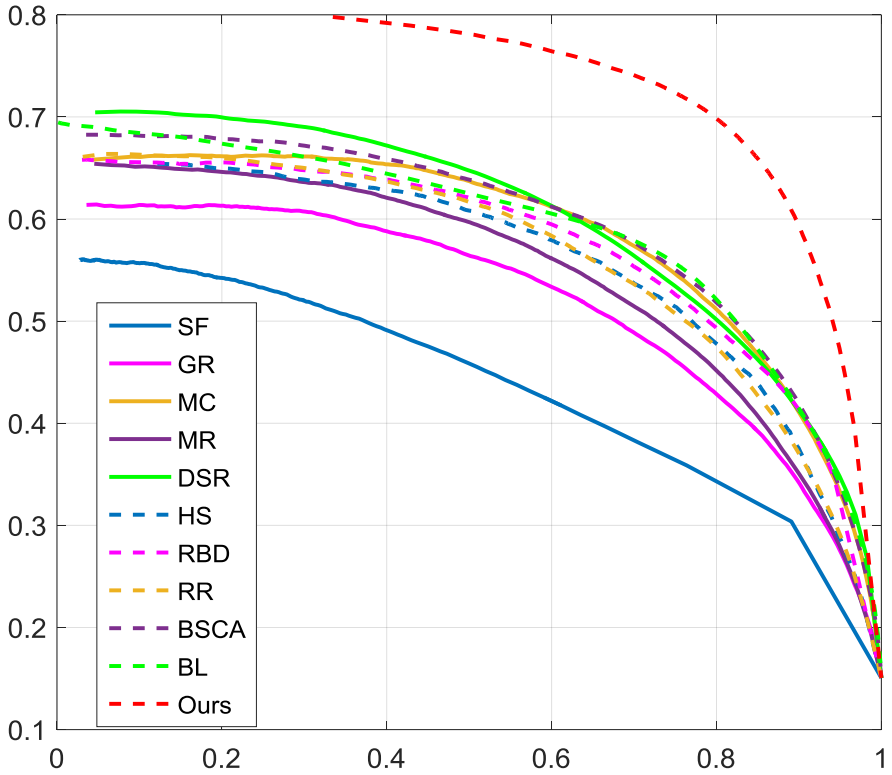

Figure 5.29 Precision-recall curves against conventional methods on the HKU-IS dataset.

Table 5.11 Quantitative evaluation results of DSL against conventional saliency detection methods

Dataset	ECSSD		PASCAL-S		SED1		SED2		THUR15K		HKU-IS	
Metric	F-Measure	MAE	F-Measure	MAE	F-Measure	MAE	F-Measure	MAE	F-Measure	MAE	F-Measure	MAE
SF	0.549	0.268	0.496	0.241	0.665	0.234	0.783	0.171	0.469	0.193	0.588	0.183
GR	0.642	0.317	0.604	0.301	0.791	0.224	0.785	0.192	0.551	0.264	0.672	0.266
MC	0.703	0.251	**0.668**	0.232	**0.844**	0.164	0.775	0.180	**0.610**	0.199	0.723	0.201
MR	0.708	0.236	0.612	0.259	0.841	**0.143**	0.771	0.164	0.573	0.209	0.689	0.192
DSR	0.699	**0.226**	0.651	**0.208**	0.819	0.160	**0.793**	**0.140**	**0.611**	**0.139**	**0.735**	**0.133**
HS	0.698	0.269	0.645	0.264	0.825	0.163	0.791	0.195	0.585	0.250	0.706	0.253
RBD	0.686	**0.225**	0.659	**0.202**	0.829	0.144	**0.826**	**0.130**	0.596	**0.163**	**0.725**	**0.150**
RR	0.710	0.234	0.639	0.232	**0.843**	**0.141**	0.769	0.161	0.590	0.185	0.711	0.175
BSCA	**0.718**	0.233	**0.669**	0.224	0.832	0.155	0.780	0.158	0.609	0.216	0.722	0.210
BL	**0.716**	0.262	0.663	0.249	0.840	0.190	0.787	0.189	0.606	0.261	0.716	0.257
Ours	**0.808**	**0.126**	**0.791**	**0.122**	**0.901**	**0.099**	**0.858**	**0.108**	**0.730**	**0.123**	**0.858**	**0.125**

*For each row, the top 3 results are marked in **red**, **blue** and **green**, respectively.*

5.7.9 Evaluation Against Learning Based Methods

Since DSL is learning based, it is not surprising that it achieves large performance improvements against the conventional saliency detection methods in section 5.7.8. To further validate the effectiveness of DSL, we compare it against six state-of-the-art learning based methods, namely DRFI [57], HDCT [89], MCDL [91], LEGS [90], MDF [92] and DISC [93]. All of the six methods are published after 2013, and the last four methods are recently published in 2015. The experiments are conducted on the same six datasets in section 5.7.8, and the comparison results are shown in Figure 5.30 to Figure 5.35, as well as Table 5.12.

It is observed that the overall performances of the learning based methods are significantly higher than those of the conventional methods in Table 5.11, which is mainly attributed to the high-level features involved in their learning processes. Nevertheless, DSL still maintains remarkable advantages against the comparison learning based methods. It achieves the optimal performance on five out of six F-measures and three out of six MAEs, and achieves the second best on all of the other evaluations with close distance to the optimal. We note that MDF is the only method that uses the training set of HKU-IS (3,000 images) in the training process, so its relatively higher performance on the test set of HKU-IS is expected; nevertheless, DSL behaves closely against MDF in F-measure, and even achieves significantly better MAE. We attribute the high performance of DSL to its combination of dense and sparse labeling that exploits both macro object contours and local low-level image features. DSL's superior performance against the state-of-the-art learning based methods further validates its effectiveness and robustness in various scenarios.

To demonstrate the greatly improved performance of DSL more straightforwardly, we select typical saliency map examples from both the conventional methods and the learning based methods, which are assembled together in Figure 5.36. We note that DSL exhibits high accuracy and robustness on various challenging scenarios, including images with low contrast objects (Figure 5.36(a) - Figure 5.36(c)), images with complex foreground / background patterns (Figure 5.36(d) - Figure 5.36(f)), and images with highly interfering backgrounds (Figure 5.36(g) - Figure 5.36(h)).

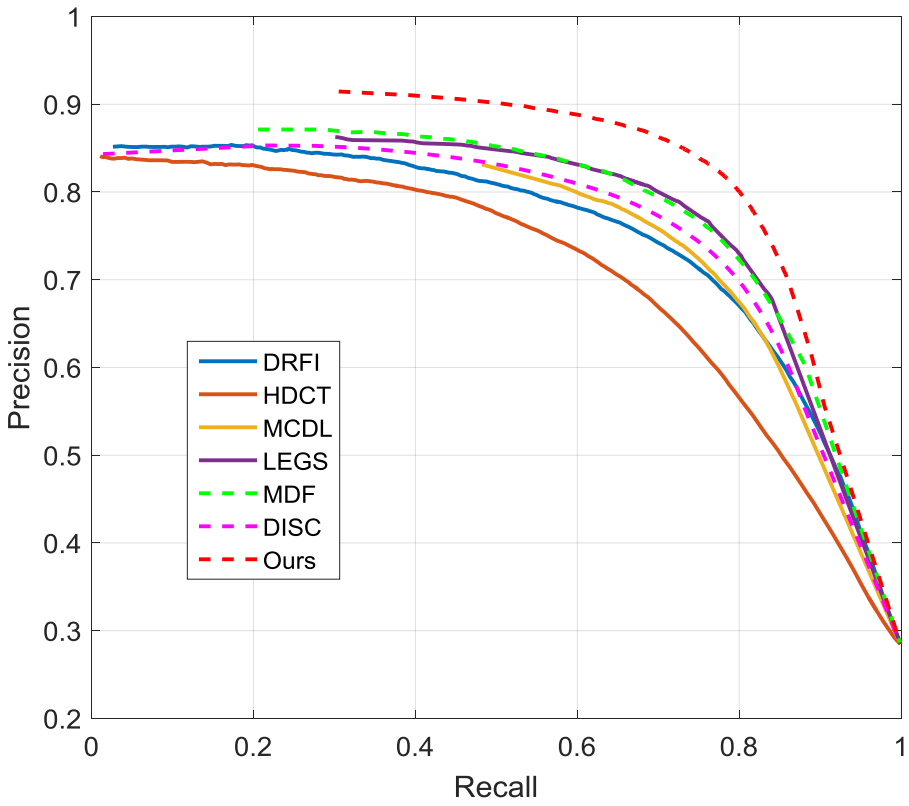

Figure 5.30 Precision-recall curves against learning based methods on the ECSSD dataset.

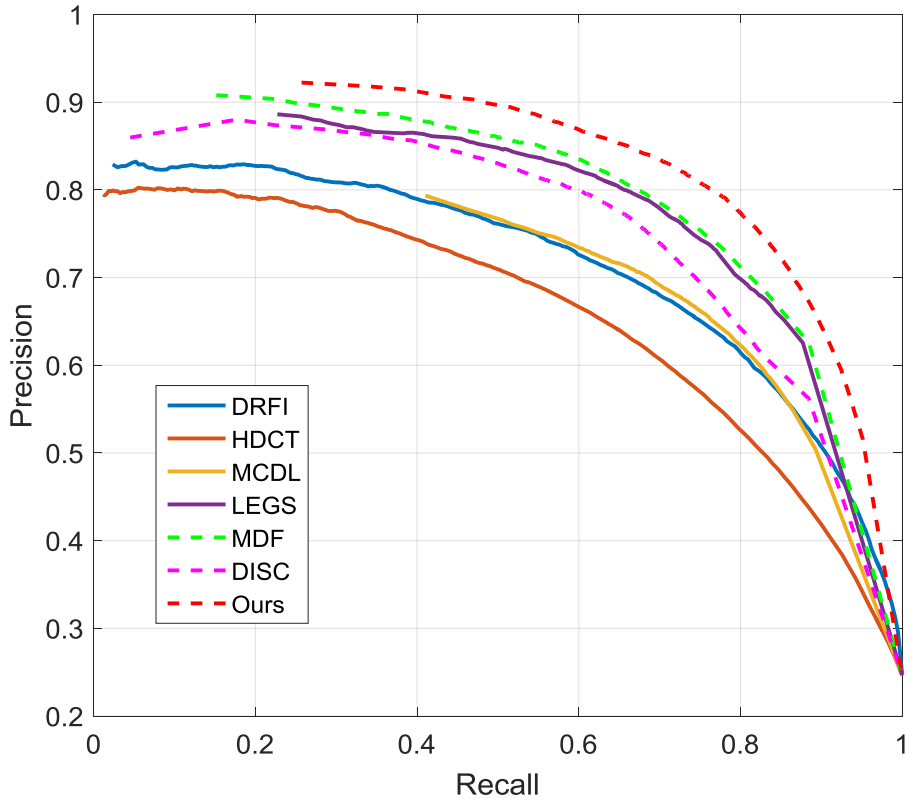

Figure 5.31 Precision-recall curves against learning based methods on the PASCAL-S dataset.

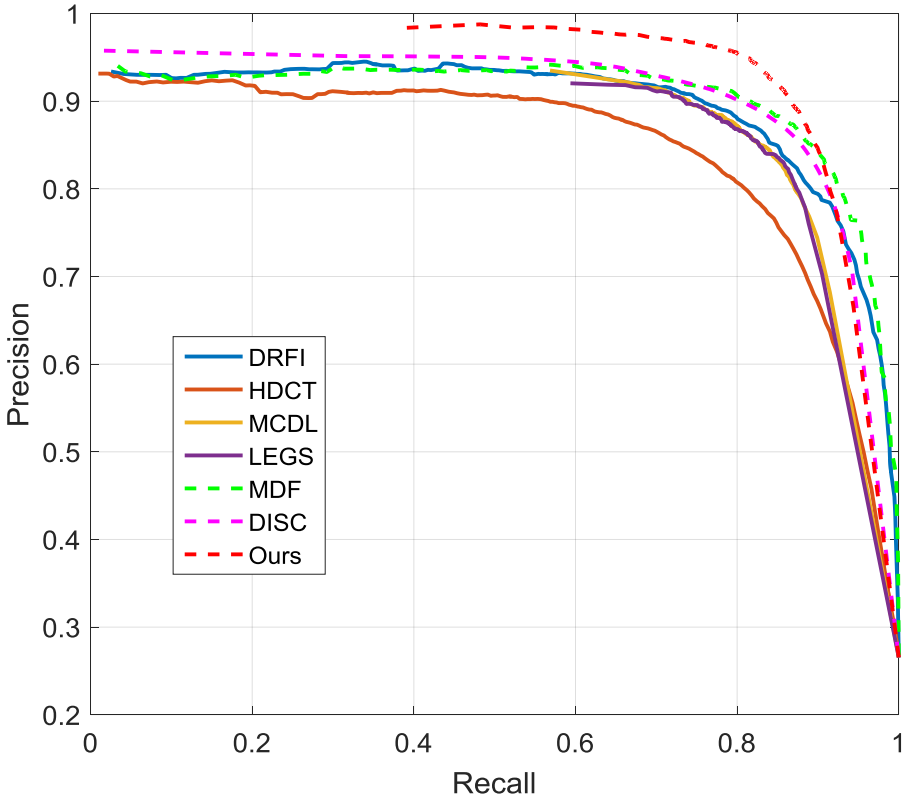

Figure 5.32 Precision-recall curves against learning based methods on the SED1 dataset.

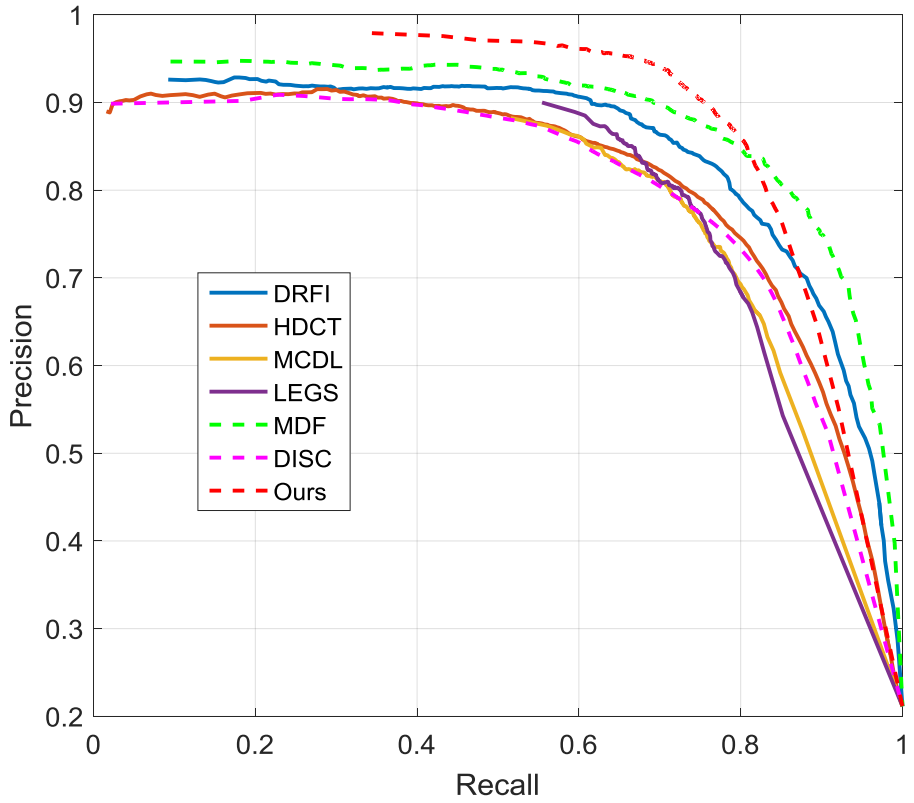

Figure 5.33 Precision-recall curves against learning based methods on the SED2 dataset.

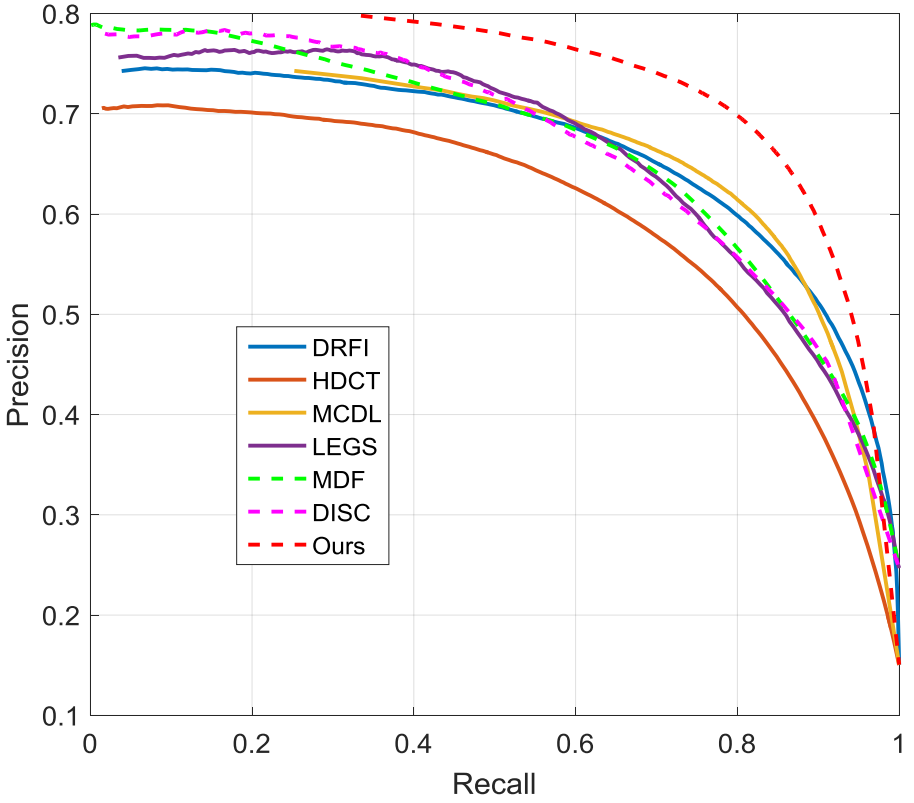

Figure 5.34 Precision-recall curves against learning based methods on the THUR15K dataset.

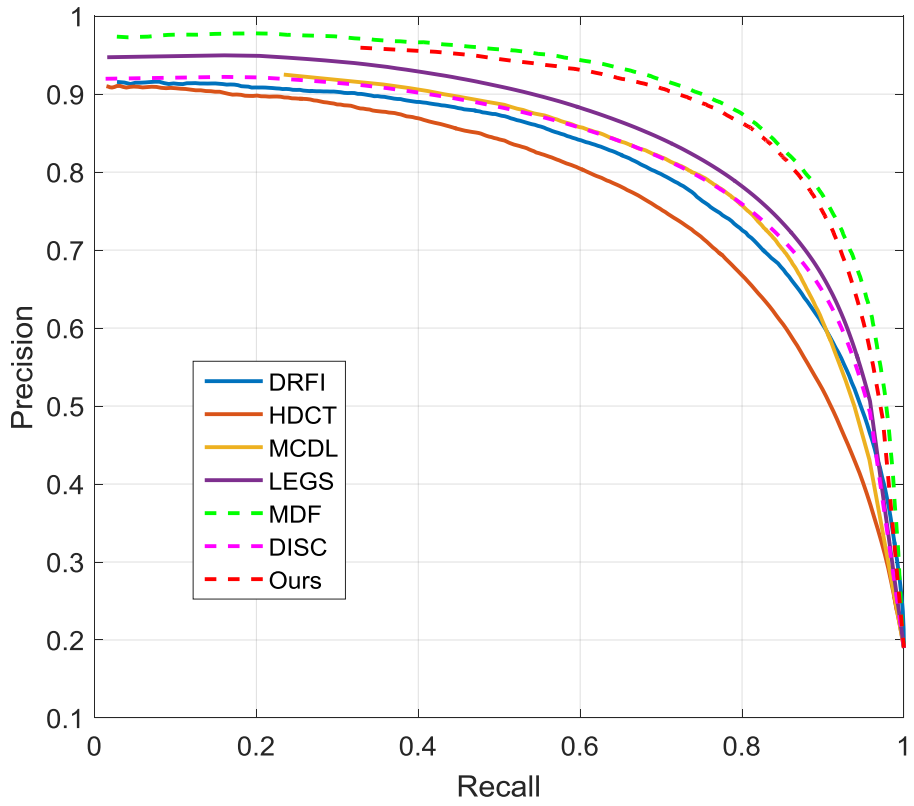

Figure 5.35 Precision-recall curves against learning based methods on the HKU-IS dataset.

Table 5.12 Quantitative evaluation results of DSL against learning based saliency detection methods

Dataset	ECSSD		PASCAL-S		SED1		SED2		THUR15K		HKU-IS	
Metric	F-Measure	MAE	F-Measure	MAE	F-Measure	MAE	F-Measure	MAE	F-Measure	MAE	F-Measure	MAE
DRFI	0.736	0.226	0.694	0.210	0.864	0.149	**0.823**	0.140	0.666	0.169	0.775	0.161
HDCT	0.698	**0.166**	0.652	**0.157**	0.821	0.183	0.792	**0.134**	0.620	0.163	0.747	**0.155**
MCDL	0.748	0.175	0.700	0.160	0.858	**0.087**	0.785	0.137	0.673	0.192	0.789	0.181
LEGS	**0.776**	0.182	**0.762**	0.171	0.867	0.185	0.802	**0.104**	**0.688**	0.155	**0.837**	**0.146**
MDF	**0.772**	**0.174**	**0.768**	**0.144**	**0.881**	0.158	**0.844**	0.152	**0.701**	**0.140**	**0.860**	0.209
DISC	0.756	0.208	0.744	0.172	**0.876**	**0.118**	0.780	0.153	0.664	**0.084**	0.788	0.180
Ours	**0.808**	**0.126**	**0.791**	**0.122**	**0.901**	**0.099**	**0.858**	**0.108**	**0.730**	**0.123**	**0.858**	**0.125**

*For each row, the top 3 results are marked in **red**, **blue** and **green**, respectively.*

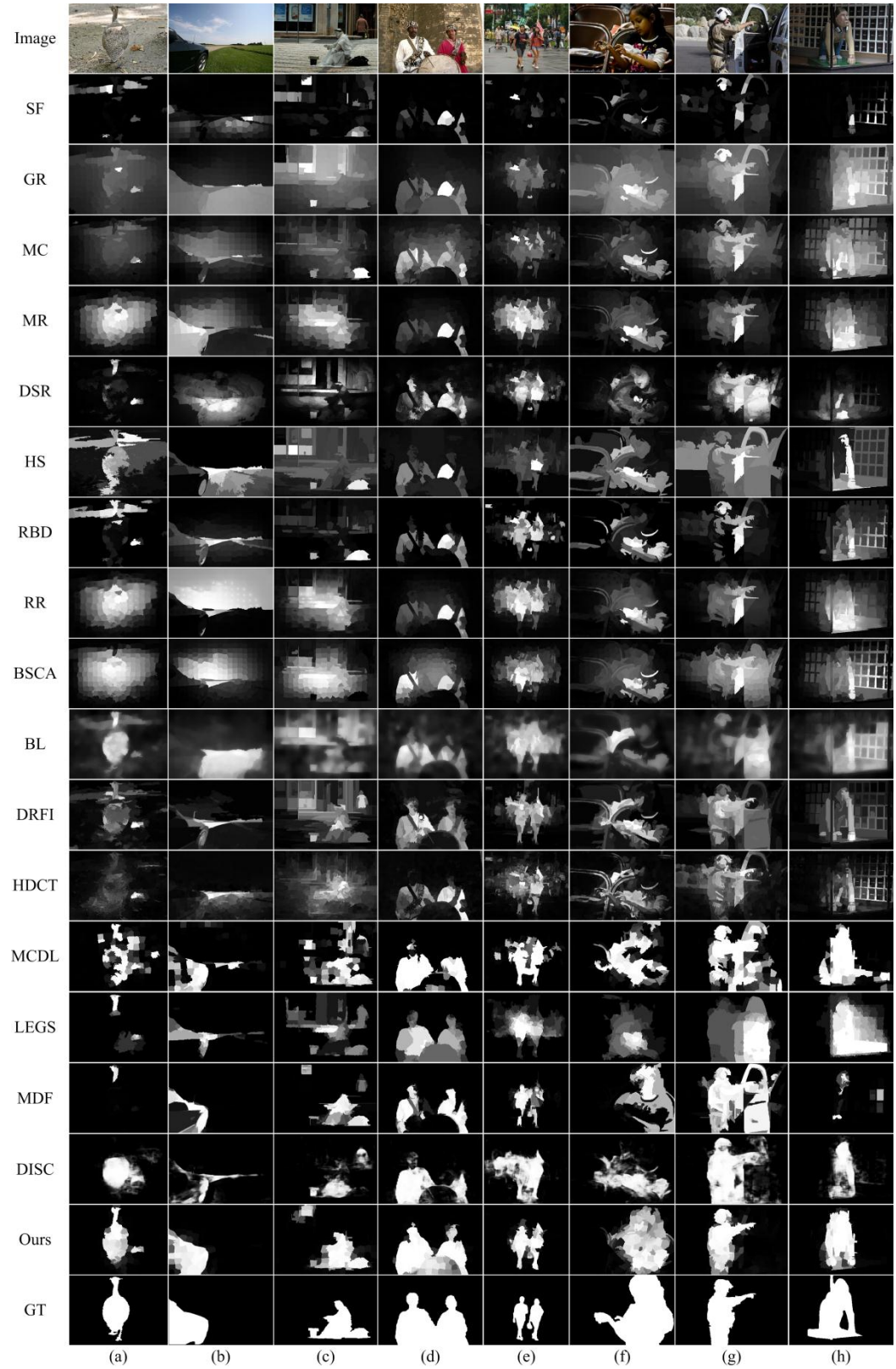

Figure 5.36 Saliency map examples of state-of-the-art methods against our DSL method. (a) – (c): images with low contrast objects; (d) – (f): images with complex foreground / background patterns; (g) - (h): images with highly interfering background.

5.7.10 Efficiency

To evaluate the efficiency of DSL, we select two comparison methods from both the conventional methods and the learning based methods that have the highest performances in Table 5.11 and Table 5.12, namely DSR, RBD, LEGS and MDF. We record their average running time per image on the same machine described in section 5.7.3, and list the results in Table 5.13. Since all of the five methods are implemented in MATLAB, the efficiency comparison is fair in terms of coding language. It is seen that besides its premium performances against the comparison methods, DSL also achieves comparable efficiency to the conventional methods, and notably faster speed than the learning based methods. The three steps of DL, SL and DC take approximately 5%, 60% and 35% of the total running time, respectively.

Table 5.13 Efficiency comparison (seconds per image)

Method	**DSR**	**RBD**	**LEGS**	**MDF**	**DSL**
Time (s)	0.525	0.341	1.75	1.48	0.695
Code	MATLAB	MATLAB	MATLAB	MATLAB	MATLAB

5.7.11 Limitation

As mentioned in section 5.7.8, currently DSL's high performance is only guaranteed on single-object images, which is mainly due to the single-object training set we used to train the DL, SL and DC networks. This issue, however, is an inherent limitation with all of the learning based methods that depend on the training data. We can solve this issue by extending our training set with broader categories of images, which will be covered in our future works.

5.8 Summary

In this chapter, we have proposed two DNN-based saliency detection methods, namely BSFE and DSL.

The BSFE method is based on stacked auto-encoder (SAE); compared to most existing methods which simply treat image boundaries as background query seeds, BSFE self-adaptively searches background via the proposed BS SAE model. The saliency map is then produced by the following FE SAE model, which hierarchically utilizes the capacity of data reconstruction of AE. BSFE is compared against six popular state-of-the-art methods on four datasets, the results of which demonstrate its favorable performance both quantitatively and qualitatively.

On the other hand, the DSL method conducts dense and sparse labeling of image saliency with multi-dimensional features. DSL consists of three major steps, namely DL, SL and DC. The DL and SL steps conduct effective initial saliency estimations with both macro object contours and local low-level features, while the final DC network establishes a 6-channeled data structure as input, and conducts accurate final saliency classification. Our DSL method achieves remarkably higher performance against sixteen state-of-the-art saliency detection methods (including ten conventional methods and six learning based methods) on six well-recognized public datasets, in terms of both accuracy and robustness. Besides that, DSL also maintains its efficiency in the same level of conventional methods, and behaves significantly faster than the other learning based methods.

Chapter 6 Conclusions and Future Works

6.1 Conclusions

This thesis focuses on image saliency detection, which is an important task in computer vision. Three main parts of content have been presented.

In the first part (Chapter 1), we have introduced image saliency detection as a computer vision problem, including its history of development, significance in both academia and industry, and the challenges face by existing methods.

In the second part (Chapter 2), we have systematically reviewed the related works to this thesis, including saliency detection, image segmentation, object proposal generation and deep neural network (DNN). Specifically, in the review of saliency detection, various state-of-the-art bottom-up, top-down and unconventional saliency detection methods have been introduced; while in the review of DNN, we have illustrated its fundamental principles, as well as its applications in sparse labeling and dense labeling.

In the third part (Chapter 3 to Chapter 5), which is the major part of this thesis, we have proposed four novel saliency detection methods in two categories, namely conventional low-level feature based saliency detection methods, and DNN based saliency detection methods:

(1) In Chapter 3, we have introduced the RR method, which is based on conventional hand crafted low-level image features. It first filters out one of the four boundaries of the input image that most unlikely belong to the background, effectively neutralizes the

negative influences of boundary-adjacent foreground regions in the saliency estimations. The regularized random walks ranking (RRWR) algorithm, which is based on the Dirichlet function and has a newly proposed fitting constraint, is then conducted to generate pixel-wised saliency maps that reflect full-details of the input image.

(2) In Chapter 4, we have introduced the RCRR method, which is an improved version of the RR method that involves the reversion correction (RC) process to better refine the image boundaries. Instead of completely removing one of the problematic boundaries, the RC process locates and eliminates the boundary-adjacent foreground superpixels, which is more accurate and can maximally prevent the saliency reversions from emerging. We also present the extensibility our method as a saliency optimization algorithm, which can be directly applied on existing saliency detection methods for performance improvement purposes. Besides that, we propose the boundary-adjacent object saliency (BAOS) dataset, which is a 200-image dataset that provides an objective evaluation for saliency detection methods' performance on boundary-adjacent salient objects.

(3) In Chapter 5, we have introduced the BSFE method, which is based on stacked auto-encoder (SAE). Compared to most existing methods which simply treat image boundaries as background query seeds, BSFE self-adaptively searches background via the proposed BS SAE model. The saliency map is then produced by the following FE SAE model, which hierarchically utilizes the capacity of data reconstruction of AE.

(4) In Chapter 5, we also introduced the DSL method, which is based on multiple convolutional neural networks (CNNs) and multi-dimensional features. DSL consists of three major steps, namely DL, SL and DC. The DL and SL steps conduct effective initial saliency estimations with both macro object contours and local low-level features, while

the final DC network establishes a 6-channeled data structure as input, and conducts accurate final saliency classification.

All of the four methods behave favorably against state-of-the-art saliency detection methods in their experimental evaluations, especially the DSL method, which achieves remarkably higher performance against sixteen state-of-the-art saliency detection methods (including ten conventional methods and six learning based methods) on six well-recognized public datasets, in terms of both accuracy and robustness.

6.2 Future Works

The successes of our proposed methods demonstrate the combined strength of low-level image features and DNNs in saliency detection, and also illustrate more potential applications of saliency detection in computer vision tasks.

In the future, we will focus on addressing the limitations in our existing methods, as well as exploring for new and even better models in saliency detection. For example, the RC algorithm in section 4 still encounters difficulty when dealing images with large portion of boundaries covered by the foreground object, and a better low-level feature based method that can extract foreground / background queries beyond the constraint of image boundaries is desired. We will also establish new network frameworks that can better utilize the dense and sparse labeling capacities of DNN, as well as enriching the training dataset, so that more categories of image cases can be covered.

Moreover, we will further explore for new adaptations of our existing methods in more challenging computer vision tasks, such as the applications in part-based object detection [189], [190], fine-grained image classification [191], [192], medical image segmentation [69], [193] and video data processing [194], [195].

We believe that with the refinements of our proposed methods and the explorations of new potential applications, the general task of visual saliency detection will be better understood and solved, which will further facilitate other related tasks in computer vision, and create more value to the future.

References

[1] C. Li, Y. Yuan, W. Cai, Y. Xia, and D. D. Feng, "Robust saliency detection via regularized random walks ranking," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 2710-2717.

[2] Y. Yuan, C. Li, J. Kim, W. Cai, and D. D. Feng, "Reversion correction and regularized random walks ranking for saliency detection," *IEEE Trans. Image Processing*, 2016.

[3] K. Yan, C. Li, X. Wang, A. Li, Y. Yuan, J. Kim, *et al.*, "Adaptive background search and foreground estimation for saliency detection via comprehensive autoencoder," in *IEEE Int. Conf. Image Process. (ICIP)*, Phoenix, AZ, USA, Sep. 2016, pp. 2767-2771.

[4] Y. Yuan, C. Li, J. Kim, W. Cai, and D. D. Feng, "Dense and sparse labeling with multi-dimensional features for saliency detection," *IEEE Trans. Circuits Syst. Video Techn.*, vol. PP, no. 99, pp. 1-14, 2016.

[5] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254-1259, Nov. 1998.

[6] L. Itti and C. Koch, "Computational modelling of visual attention," *Nature Reviews Neuroscience*, vol. 2, no. 3, pp. 194-203, 2001.

[7] Y.-F. Ma and H.-J. Zhang, "Contrast-based image attention analysis by using fuzzy growing," in *Proc. ACM Multimedia (ACMMM)*, Berkeley, CA, USA, Nov. 2003, pp. 374-381.

[8] A. Borji, D. N. Sihite, and L. Itti, "Salient object detection: A benchmark," in *Comput. Vision–ECCV*, ed: Springer, 2012, pp. 414-429.

[9] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, Miami, FL, USA, Jun. 2009, pp. 2106-2113.

[10] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, *et al.*, "Learning to detect a salient object," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 2, pp. 353-367, 2011.

[11] C. Shen and Q. Zhao, "Learning to predict eye fixations for semantic contents using multi-layer sparse network," *Neurocomputing*, vol. 138, pp. 61-68, 2014.

[12] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167-181, 2004.

[13] U. Rutishauser, D. Walther, C. Koch, and P. Perona, "Is bottom-up attention useful for object recognition?," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, Washington, DC, USA, Jun. 2004, pp. II-37-II-44 Vol. 2.

[14] C. Kanan and G. Cottrell, "Robust classification of objects, faces, and flowers using natural image statistics," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, San Francisco, CA, USA, Jun. 2010, pp. 2472-2479.

[15] F. Moosmann, D. Larlus, and F. Jurie, "Learning saliency maps for object categorization," in *Int. Workshop Representation Use Prior Knowledge Vision*, Graz, Austria, May. 2006.

[16] H. Shen, S. Li, C. Zhu, H. Chang, and J. Zhang, "Moving object detection in aerial video based on spatiotemporal saliency," *Chinese J. Aeronautics*, vol. 26, no. 5, pp. 1211-1217, 2013.

[17] Z. Ren, S. Gao, L.-T. Chia, and I. W.-H. Tsang, "Region-based saliency detection and its application in object recognition," *IEEE Trans. Circuits Syst. Video Techn.*, vol. 24, no. 5, pp. 769-779, 2014.

[18] A. Borji, M. N. Ahmadabadi, and B. N. Araabi, "Cost-sensitive learning of top-down modulation for attentional control," *Mach. Vision Applicat.*, vol. 22, no. 1, pp. 61-76, 2011.

[19] A. Borji and L. Itti, "Scene classification with a sparse set of salient regions," in *IEEE Int. Conf. Robotics Automation (ICRA)*, Shanghai, China, May. 2011, pp. 1902-1908.

[20] A. Borji, M.-M. Cheng, H. Jiang, and J. Li, "Salient object detection: A survey," *arXiv preprint arXiv:1411.5878*, 2014.

[21] A. Karpathy, S. Miller, and L. Fei-Fei, "Object discovery in 3d scenes via shape analysis," in *IEEE Int. Conf. Robotics Automation (ICRA)*, Karlsruhe, Germany, May. 2013, pp. 2088-2095.

[22] S. Frintrop, G. M. Garc á, and A. B. Cremers, "A Cognitive Approach for Object Discovery," in *Int. Conf. Pattern Recognition (ICPR)*, Stockholm, Sweden, Aug. 2014, pp. 2329-2334.

[23] J. Wang, L. Quan, J. Sun, X. Tang, and H.-Y. Shum, "Picture collage," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, New York City, NY, USA, Jun. 2006, pp. 347-354.

[24] S. Goferman, A. Tal, and L. Zelnik‐Manor, "Puzzle‐like Collage," in *Comput. Graph. Forum.* 2010, pp. 459-468.

[25] H. Huang, L. Zhang, and H.-C. Zhang, "Arcimboldo-like collage using internet images," in *ACM Trans. Graph. (TOG)* 2011, p. 155.

[26] A. Ninassi, O. Le Meur, P. Le Callet, and D. Barba, "Does where you gaze on an image affect your perception of quality? Applying visual attention to image quality metric," in *IEEE Int. Conf. Image Process. (ICIP)*, San Antonio, TX, USA, Sep. 2007, pp. II-169-II-172.

[27] H. Liu and I. Heynderickx, "Studying the added value of visual attention in objective image quality metrics based on eye movement data," in *IEEE Int. Conf. Image Process. (ICIP)*, Cairo, Egypt, Nov. 2009, pp. 3097-3100.

[28] A. Li, X. She, and Q. Sun, "Color image quality assessment combining saliency and fsim," in *Int. Conf. Digital Image Process. (ICDIP)*, Beijing, China, Apr. 2013, pp. 88780I-88780I-5.

[29] M. Donoser, M. Urschler, M. Hirzer, and H. Bischof, "Saliency driven total variation segmentation," in *IEEE Int. Conf. Comput. Vision (ICCV)*, Kyoto, Japan, Sep. 2009, pp. 817-824.

[30] A. Y.-S. Chia, S. Zhuo, R. K. Gupta, Y.-W. Tai, S.-Y. Cho, P. Tan, *et al.*, "Semantic colorization with internet images," in *ACM Trans. Graph. (TOG)* 2011, p. 156.

[31] C. Qin, G. Zhang, Y. Zhou, W. Tao, and Z. Cao, "Integration of the saliency-based seed extraction and random walks for image segmentation," *Neurocomputing*, vol. 129, pp. 378-391, 2014.

[32] M. Johnson-Roberson, J. Bohg, M. Björkman, and D. Kragic, "Attention-based active 3D point cloud segmentation," in *IEEE/RSJ Int. Conf. Intelligent Robots Syst. (IROS)*, Taipei, Taiwan, Oct. 2010, pp. 1165-1170.

[33] T. Chen, M.-M. Cheng, P. Tan, A. Shamir, and S.-M. Hu, "Sketch2Photo: internet image montage," *ACM Trans. Graph. (TOG)* 2009, vol. 28, no. 5, p. 124.

[34] S. Feng, D. Xu, and X. Yang, "Attention-driven salient edge (s) and region (s) extraction with application to CBIR," *Signal Process.*, vol. 90, no. 1, pp. 1-15, 2010.

[35] J. Sun, J. Xie, J. Liu, and T. Sikora, "Image adaptation and dynamic browsing based on two-layer saliency combination," *IEEE Trans. Broadcast.*, vol. 59, no. 4, pp. 602-613, 2013.

[36] H. Liu, L. Zhang, and H. Huang, "Web-image driven best views of 3d shapes," *Visual Comput.*, vol. 28, no. 3, pp. 279-287, 2012.

[37] R. Margolin, L. Zelnik-Manor, and A. Tal, "Saliency for image manipulation," *Visual Comput.*, vol. 29, no. 5, pp. 381-392, 2013.

[38] C. Goldberg, T. Chen, F. L. Zhang, A. Shamir, and S. M. Hu, "Data‐Driven Object Manipulation in Images," in *Comput. Graph. Forum*, 2012, pp. 265-274.

[39] C. Guo and L. Zhang, "A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression," *IEEE Trans. Image Process.*, vol. 19, no. 1, pp. 185-198, 2010.

[40] L. Itti, "Automatic foveation for video compression using a neurobiological model of visual attention," *IEEE Trans. Image Process.*, vol. 13, no. 10, pp. 1304-1318, 2004.

[41] Y.-F. Ma, X.-S. Hua, L. Lu, and H.-J. Zhang, "A generic framework of user attention model and its application in video summarization," *IEEE Trans. Multimedia*, vol. 7, no. 5, pp. 907-919, 2005.

[42] Y. J. Lee, J. Ghosh, and K. Grauman, "Discovering important people and objects for egocentric video summarization," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, Providence, RI, USA, Jun. 2012, p. 7.

[43] Q.-G. Ji, Z.-D. Fang, Z.-H. Xie, and Z.-M. Lu, "Video abstraction based on the visual attention model and online clustering," *Signal Process. Image Commun.*, vol. 28, no. 3, pp. 241-253, 2013.

[44] S. Stalder, H. Grabner, and L. Van Gool, "Dynamic objectness for adaptive tracking," in *Asian Conf. Comput. Vision (ACCV)*, Daejeon, Korea, Nov. 2012, pp. 43-56.

[45] G. M. Garc á, D. A. Klein, J. St ückler, S. Frintrop, and A. B. Cremers, "Adaptive multi-cue 3D tracking of arbitrary objects," in *Joint DAGM OAGM Symp. (DAGM-OAGM)*, Graz, Austria, Aug. 2012, pp. 357-366.

[46] A. Borji, S. Frintrop, D. N. Sihite, and L. Itti, "Adaptive object tracking by learning background context," in *Workshop IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, Providence, RI, USA, Jun. 2012, pp. 23-30.

[47] D. A. Klein, D. Schulz, S. Frintrop, and A. B. Cremers, "Adaptive real-time video-tracking for arbitrary objects," in *IEEE/RSJ Int. Conf. Intelligent Robots Syst. (IROS)*, Taipei, Taiwan, Oct. 2010, pp. 772-777.

[48] S. Frintrop and M. Kessel, "Most salient region tracking," in *IEEE Int. Conf. Robotics Automation (ICRA)*, Kobe, Japan, May. 2009, pp. 1869-1874.

[49] G. Zhang, Z. Yuan, N. Zheng, X. Sheng, and T. Liu, "Visual saliency based object tracking," in *Asian Conf. Comput. Vision (ACCV)*, Xi'an, China, Sep. 2009, pp. 193-203.

[50] D. Meger, P.-E. Forssén, K. Lai, S. Helmer, S. McCann, T. Southey, *et al.*, "Curious george: An attentive semantic robot," *Robotics Autonomous Syst.*, vol. 56, no. 6, pp. 503-511, 2008.

[51] Y. Sugano, Y. Matsushita, and Y. Sato, "Calibration-free gaze sensing using saliency maps," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, San Francisco, CA, USA, Jun. 2010, pp. 2667-2674.

[52] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, "Saliency detection via graph-based manifold ranking," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, Portland, OR, USA, Jun. 2013, pp. 3166-3173.

[53] W. Zhu, S. Liang, Y. Wei, and J. Sun, "Saliency Optimization from Robust Background Detection," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, Columbus, OH, USA, Jun. 2014, pp. 2814-2821.

[54] B. Jiang, L. Zhang, H. Lu, C. Yang, and M.-H. Yang, "Saliency detection via absorbing markov chain," in *IEEE Int. Conf. Comput. Vision (ICCV)*, Sydney, Australia, Dec. 2013, pp. 1665-1672.

[55] D. A. Klein and S. Frintrop, "Center-surround divergence of feature statistics for salient object detection," in *IEEE Int. Conf. Comput. Vision (ICCV)*, Barcelona, Spain, Nov. 2011, pp. 2214-2219.

[56] D. G. Lowe, "Object recognition from local scale-invariant features," in *IEEE Int. Conf. Comput. Vision (ICCV)*, Kerkyra, Greece, Sep. 1999, pp. 1150-1157.

[57] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li, "Salient object detection: A discriminative regional feature integration approach," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, Portland, OR, USA, Jun. 2013, pp. 2083-2090.

[58] K. Fu, C. Gong, J. Yang, Y. Zhou, and I. Yu-Hua Gu, "Superpixel based color contrast and color distribution driven salient object detection," *Signal Process. Image Commun.*, vol. 28, no. 10, pp. 1448-1463, Jul. 2013.

[59] L. Deng and D. Yu, "Deep learning: methods and applications," *Found. Trends Signal Process.*, vol. 7, no. 3–4, pp. 197-387, 2014.

[60] J. Sun, H. Lu, and X. Liu, "Saliency Region Detection Based on Markov Absorption Probabilities," *IEEE Trans. Image Process.*, vol. 24, no. 5, pp. 1639-1649, 2015.

[61] E. Rahtu, J. Kannala, M. Salo, and J. Heikkilä, "Segmenting salient objects from images and videos," in *Comput. Vision–ECCV*, ed: Springer, 2010, pp. 366-379.

[62] N. Bruce and J. Tsotsos, "Saliency based on information maximization," in *Advances Neural Inform. Process. Syst.* 2005, pp. 155-162.

[63] M.-M. Cheng, G.-X. Zhang, N. J. Mitra, X. Huang, and S.-M. Hu, "Global contrast based salient region detection," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, Colorado Springs, CO, USA, Jun. 2011, pp. 409-416.

[64] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S. Hu, "Global contrast based salient region detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 569-582, May. 2015.

[65] L. Duan, C. Wu, J. Miao, L. Qing, and Y. Fu, "Visual saliency detection by spatially weighted dissimilarity," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, Colorado Springs, CO, USA, Jun. 2011, pp. 473-480.

[66] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, Miami, FL, USA, Jun. 2009, pp. 1597-1604.

[67] X. Hou and L. Zhang, "Saliency detection: A spectral residual approach," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, Minneapolis, MN, USA, Jun. 2007, pp. 1-8.

[68] C. Guo, Q. Ma, and L. Zhang, "Spatio-temporal saliency detection using phase spectrum of quaternion fourier transform," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, Anchorage, AL, USA, Jun. 2008, pp. 1-8.

[69] A. Li, C. Li, X. Wang, S. Eberl, D. D. Feng, and M. Fulham, "Automated Segmentation of Prostate MR Images Using Prior Knowledge Enhanced Random Walker," in *Int. Conf. Digital Computing: Tech. Applicat. (DICTA)*, Hobart, Australia, Nov. 2013, pp. 1-7.

[70] F. Perazzi, P. Krahenbuhl, Y. Pritch, and A. Hornung, "Saliency filters: Contrast based filtering for salient region detection," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, Providence, RI, USA, Jun. 2012, pp. 733-740.

[71] Y. Wei, F. Wen, W. Zhu, and J. Sun, "Geodesic saliency using background priors," in *Comput. Vision–ECCV*, ed: Springer, 2012, pp. 29-42.

[72] Y. Qin, H. Lu, Y. Xu, and H. Wang, "Saliency detection via cellular automata," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 110-119.

[73] L. Zhang, M. H. Tong, T. K. Marks, H. Shan, and G. W. Cottrell, "SUN: A Bayesian framework for saliency using natural statistics," *J. Vision*, vol. 8, no. 7, p. 32, 2008.

[74] K.-Y. Chang, T.-L. Liu, H.-T. Chen, and S.-H. Lai, "Fusing generic objectness and visual saliency for salient object detection," in *IEEE Int. Conf. Comput. Vision (ICCV)*, Barcelona, Spain, Nov. 2011, pp. 914-921.

[75] H. Jiang, J. Wang, Z. Yuan, T. Liu, N. Zheng, and S. Li, "Automatic salient object segmentation based on context and shape prior," in *British Mach. Vision Conf. (BMVC)*, Dundee, UK, Aug. 2011, p. 7.

[76] H. R. Tavakoli, E. Rahtu, and J. Heikkilä, "Fast and efficient saliency detection using sparse sampling and kernel density estimation," in *Image Analysis*, ed: Springer, 2011, pp. 666-675.

[77] X. Shen and Y. Wu, "A unified approach to salient object detection via low rank matrix recovery," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, Providence, RI, USA, Jun. 2012, pp. 853-860.

[78] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 10, pp. 1915-1926, Oct. 2012.

[79] R. Margolin, A. Tal, and L. Zelnik-Manor, "What Makes a Patch Distinct?," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, Portland, OR, USA, Jun. 2013, pp. 1139-1146.

[80] X. Li, H. Lu, L. Zhang, X. Ruan, and M.-H. Yang, "Saliency detection via dense and sparse reconstruction," in *IEEE Int. Conf. Comput. Vision (ICCV)*, Sydney, Australia, Dec. 2013, pp. 2976-2983.

[81] 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 Int. Conf. Comput. Vision (ICCV)*, Sydney, Australia, Dec. 2013, pp. 1529-1536.

[82] P. Jiang, H. Ling, J. Yu, and J. Peng, "Salient region detection by ufo: Uniqueness, focusness and objectness," in *IEEE Int. Conf. Comput. Vision (ICCV)*, Sydney, Australia, Dec. 2013, pp. 1976-1983.

[83] C. Yang, L. Zhang, and H. Lu, "Graph-regularized saliency detection with convex-hull-based center prior," *IEEE Signal Process. Lett.*, vol. 20, no. 7, pp. 637-640, 2013.

[84] J. Yang and M.-H. Yang, "Top-down visual saliency via joint crf and dictionary learning," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, Providence, RI, USA, Jun. 2012, pp. 2296-2303.

[85] S. Lu, V. Mahadevan, and N. Vasconcelos, "Learning Optimal Seeds for Diffusion-based Salient Object Detection," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, Portland, OR, USA, Jun. 2013, pp. 2790-2797.

[86] L. Mai, Y. Niu, and F. Liu, "Saliency aggregation: A data-driven approach," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, Portland, OR, USA, Jun. 2013, pp. 1131-1138.

[87] N. Tong, H. Lu, X. Ruan, and M.-H. Yang, "Salient object detection via bootstrap learning," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 1884-1892.

[88] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learning*, vol. 20, no. 3, pp. 273-297, 1995.

[89] J. Kim, D. Han, Y.-W. Tai, and J. Kim, "Salient region detection via high-dimensional color transform," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, Columbus, OH, USA, Jun. 2014, pp. 883-890.

[90] L. Wang, H. Lu, X. Ruan, and M.-H. Yang, "Deep networks for saliency detection via local estimation and global search," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 3183-3192.

[91] R. Zhao, W. Ouyang, H. Li, and X. Wang, "Saliency detection by multi-context deep learning," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 1265-1274.

[92] G. Li and Y. Yu, "Visual Saliency Based on Multiscale Deep Features," *arXiv preprint arXiv:1503.08663*, 2015.

[93] T. Chen, L. Lin, L. Liu, X. Luo, and X. Li, "DISC: Deep Image Saliency Computing via Progressive Representation Learning," *arXiv preprint arXiv:1511.04192*, 2015.

[94] N. Li, J. Ye, Y. Ji, H. Ling, and J. Yu, "Saliency Detection on Light Field," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, Columbus, OH, USA, Jun. 2014, pp. 2806-2813.

[95] R. Liu, J. Cao, Z. Lin, and S. Shan, "Adaptive partial differential equation learning for visual saliency detection," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, Columbus, OH, USA, Jun. 2014, pp. 3866-3873.

[96] F. Yang, H. Lu, and M.-H. Yang, "Robust superpixel tracking," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1639-1651, Apr. 2014.

[97] F. Zhou, S. B. Kang, and M. F. Cohen, "Time-mapping using space-time saliency," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, Columbus, OH, USA, Jun. 2014, pp. 3359-3365.

[98] E. Vig, M. Dorr, and D. Cox, "Large-Scale Optimization of Hierarchical Features for Saliency Prediction in Natural Images," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, Columbus, OH, USA, Jun. 2014, pp. 2798-2805.

[99] H.-D. Cheng, X. Jiang, Y. Sun, and J. Wang, "Color image segmentation: advances and prospects," *Pattern Recognition,* vol. 34, no. 12, pp. 2259-2281, 2001.

[100] N. R. Pal and S. K. Pal, "A review on image segmentation techniques," *Pattern Recognition,* vol. 26, no. 9, pp. 1277-1294, 1993.

[101] B. Peng, L. Zhang, and D. Zhang, "A survey of graph theoretical approaches to image segmentation," *Pattern Recognition,* vol. 46, no. 3, pp. 1020-1038, 2013.

[102] D. L. Pham, C. Xu, and J. L. Prince, "Current methods in medical image segmentation 1," *Annu. Review Biomedical Eng.*, vol. 2, no. 1, pp. 315-337, 2000.

[103] A. Rosenfeld and P. De La Torre, "Histogram concavity analysis as an aid in threshold selection," *IEEE Trans. Syst. Man Cybern.*, no. 2, pp. 231-235, 1983.

[104] L. G. Shapiro and G. C. Stockman, *Comput. Vision*: Prentice Hall, 2001.

[105] N. Karssemeijer, "A relaxation method for image segmentation using a spatially dependent stochastic model," *Pattern Recognition Lett.,* vol. 11, no. 1, pp. 13-23, 1990.

[106] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, no. 6, pp. 679-698, 1986.

[107] R. C. Gonzalez, *Digital Image Processing*: Pearson Education India, 2009.

[108] J. Ning, L. Zhang, D. Zhang, and C. Wu, "Interactive image segmentation by maximal similarity based region merging," *Pattern Recognition,* vol. 43, no. 2, pp. 445-456, 2010.

[109] M. E. Leventon, W. E. L. Grimson, and O. Faugeras, "Statistical shape influence in geodesic active contours," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR),* Hilton Head, SC, USA, Jun. 2000, pp. 316-323 vol.1.

[110] S.-J. Lim and Y.-S. Ho, "3-D active shape image segmentation using a scale model," in *IEEE Int. Symp. Signal Process. Inform. Technology (ISSPIT),* Vancouver, Canada, Aug. 2006, pp. 168-173.

[111] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*: Courier Corporation, 1998.

[112] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick, "Graphcut textures: image and video synthesis using graph cuts," in *ACM Trans. Graph (TOG).* 2003, pp. 277-286.

[113] P. Kohli and P. H. Torr, "Efficiently solving dynamic markov random fields using graph cuts," in *IEEE Int. Conf. Comput. Vision (ICCV),* Beijing, China, Oct. 2005, pp. 922-929.

[114] Y.-F. Pan, X. Hou, and C.-L. Liu, "Text localization in natural scene images based on conditional random field," in *Int. Conf. Document Anal. Recognition (ICDAR),* Barcelona, Spain, 2009, pp. 6-10.

[115] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph. (TOG).* 2004, vol. 23, no. 3, pp. 309-314.

[116] K. Pearson, "The problem of the random walk," *Nature*, vol. 72, no. 1865, p. 294, 1905.

[117] L. Grady, "Random walks for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, pp. 1768-1783, Nov. 2006.

[118] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269-271, 1959.

[119] R. Bellman, "On a routing problem," *DTIC Document*, 1956.

[120] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. 4, no. 2, pp. 100-107, 1968.

[121] D. B. Johnson, "Efficient algorithms for shortest paths in sparse networks," *J. ACM*, vol. 24, no. 1, pp. 1-13, 1977.

[122] L. Najman and M. Schmitt, "Watershed of a continuous function," *Signal Process.*, vol. 38, no. 1, pp. 99-112, 1994.

[123] R. L. Graham and P. Hell, "On the history of the minimum spanning tree problem," *Ann. History Computing*, vol. 7, no. 1, pp. 43-57, 1985.

[124] J. Nešetřil, E. Milková, and H. Nešetřilová, "Otakar Borůvka on minimum spanning tree problem translation of both the 1926 papers, comments, history," *Discrete Math.*, vol. 233, no. 1, pp. 3-36, 2001.

[125] J. B. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," *Proc. Amer. Math. Soc.*, vol. 7, no. 1, pp. 48-50, 1956.

[126] R. C. Prim, "Shortest Connection Networks And Some Generalizations," *Bell Syst. Tech. J.*, vol. 36, no. 6, pp. 1389-1401, 1957.

[127] D. A. Bader and G. Cong, "Fast shared-memory algorithms for computing the minimum spanning forest of sparse graphs," in *Int. Parallel Distributed Process. Symp. (IPDPS)*, Santa Fe, NM, USA, Apr. 2004, p. 39.

[128] R. Courant and D. Hilbert, *Methods Math. Physics*, vol. 1: John Wiley and Sons, 2008.

[129] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr, "BING: Binarized normed gradients for objectness estimation at 300fps," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, Columbus, OH, USA, Jun. 2014, pp. 3286-3293.

[130] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," *IEEE Trans. Pattern Anal. Mach. Intell.*. 34, no. 11, pp. 2189-2202, 2012.

[131] I. Endres and D. Hoiem, "Category-independent object proposals with diverse ranking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 2, pp. 222-234, 2014.

[132] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154-171, 2013.

[133] Z. Zhang, J. Warrell, and P. H. Torr, "Proposal generation for object detection using cascaded ranking svms," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, Colorado Springs, CO, USA, Jun. 2011, pp. 1497-1504.

[134] P. Krähenbühl and V. Koltun, "Geodesic object proposals," in *Comput. Vision–ECCV*, ed: Springer, 2014, pp. 725-739.

[135] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627-1645, 2010.

[136] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, San Diego, CA, USA, Jun. 2005, pp. 886-893.

[137] H.-L. Teuber, "Physiological psychology," *Annu. Review Psychology*, vol. 6, no. 1, pp. 267-296, 1955.

[138] R. Desimone and J. Duncan, "Neural mechanisms of selective visual attention," *Annu. Review Neuroscience*, vol. 18, no. 1, pp. 193-222, 1995.

[139] X. Wang, M. Yang, S. Zhu, and Y. Lin, "Regionlets for generic object detection," in *IEEE Int. Conf. Comput. Vision (ICCV)*, Sydney, Australia, Dec. 2013, pp. 17-24.

[140] B. Alexe, T. Deselaers, and V. Ferrari, "What is an object?," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, San Francisco, CA, USA, Jun. 2010, pp. 73-80.

[141] L. Elazary and L. Itti, "Interesting objects are visually salient," *J. Vision*, vol. 8, no. 3, pp. 3-3, 2008.

[142] A. Borji, D. N. Sihite, and L. Itti, "What stands out in a scene? A study of human explicit saliency judgment," *Vision research*, vol. 91, pp. 62-77, 2013.

[143] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, *et al.*, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541-551, 1989.

[144] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances Neural Inform. Process. Syst*. 2012, pp. 1097-1105.

[145] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *arXiv preprint arXiv:1312.6229,* 2013.

[146] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, *et al.*, "Going deeper with convolutions," *arXiv preprint arXiv:1409.4842*, 2014.

[147] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in *Advances Neural Inform. Process. Syst*. 2013, pp. 2553-2561.

[148] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, Columbus, OH, USA, Jun. 2014, pp. 580-587.

[149] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich, "Feedforward semantic segmentation with zoom-out features," *arXiv preprint arXiv:1412.0774*, 2014.

[150] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *arXiv preprint arXiv:1411.4038,* 2014.

[151] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *arXiv preprint arXiv:1412.7062,* 2014.

[152] Y. Sun, X. Wang, and X. Tang, "Deep convolutional network cascade for facial point detection," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, Portland, OR, USA, Jun. 2013, pp. 3476-3483.

[153] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, Columbus, OH, USA, Jun. 2014, pp. 1891-1898.

[154] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, Columbus, OH, USA, Jun. 2014, pp. 1653-1660.

[155] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun, "Pedestrian detection with unsupervised multi-stage feature learning," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, Portland, OR, USA, Jun. 2013, pp. 3626-3633.

[156] X. Zeng, W. Ouyang, and X. Wang, "Multi-stage contextual deep learning for pedestrian detection," in *IEEE Int. Conf. Comput. Vision (ICCV)*, Sydney, Australia, Dec. 2013, pp. 121-128.

[157] Y. Yuan, Y. Shi, C. Li, J. Kim, W. Cai, Z. Han, *et al.*, "DeepGene: an advanced cancer type classifier based on deep learning and somatic point mutations," *BMC Bioinformatics*, vol. 17, no. 17, pp. 243-256, 2016.

[158] A. Ng, J. Ngiam, C. Y. Foo, Y. Mai, C. Suen, A. Coates, *et al.*, "Unsupervised feature learning and deep learning," ed: Technical report, Stanford University, 2013.

[159] P. O. Pinheiro and R. Collobert, "From Image-level to Pixel-level Labeling with Convolutional Networks," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 1713-1721.

[160] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Comput. Vision–ECCV*, ed: Springer, 2014, pp. 818-833.

[161] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556,* 2014.

[162] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *arXiv preprint arXiv:1512.03385,* 2015.

[163] D. Zhou and B. Schölkopf, "Learning from labeled and unlabeled data using random walks," in *Pattern Recognition*, ed: Springer, 2004, pp. 237-244.

[164] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf, "Ranking on data manifolds," *Advances Neural Inform. Process. Syst.,* vol. 16, pp. 169-176, Dec. 2004.

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

[166] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 34, no. 11, pp. 2274-2282, Nov. 2012.

[167] A. Borji, M.-M. Cheng, H. Jiang, and J. Li, "Salient object detection: A benchmark," *IEEE Trans. Image Process.,* vol. 24, no. 12, pp. 5706-5722, 2015.

[168] G. Hripcsak and A. S. Rothschild, "Agreement, the f-measure, and reliability in information retrieval," *J. Amer. Medical Informatics Assoc.,* vol. 12, no. 3, pp. 296-298, May. 2005.

[169] Y. Zhang, R. Hartley, J. Mashford, and S. Burn, "Superpixels via pseudo-boolean optimization," in *IEEE Int. Conf. Comput. Vision (ICCV)*, Barcelona, Spain, Nov. 2011, pp. 1387-1394.

[170] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inform. Theory*, vol. 28, no. 2, pp. 129-137, Mar. 1982.

[171] S. Alpert, M. Galun, R. Basri, and A. Brandt, "Image segmentation by probabilistic bottom-up aggregation and cue integration," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, Minneapolis, MN, USA, Jun. 2007, pp. 1-8.

[172] Y. Li, X. Hou, C. Koch, J. Rehg, and A. Yuille, "The secrets of salient object segmentation," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, Columbus, OH, USA, Jun. 2014, pp. 4321-4328.

[173] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303-338, Sep. 2010.

[174] D. R. Martin, C. C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 5, pp. 530-549, May. 2004.

[175] J. Han, D. Zhang, X. Hu, L. Guo, J. Ren, and F. Wu, "Background prior-based salient object detection via deep reconstruction residual," *IEEE Trans. Circuits Syst. Video Techn.,* vol. 25, no. 8, pp. 1309-1321, 2015.

[176] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798-1828, 2013.

[177] J. Han, D. Zhang, S. Wen, L. Guo, T. Liu, and X. Li, "Two-stage learning to predict human eye fixations via SDAEs," *IEEE Trans. Cybern.,* vol. 46, no. 2, pp. 487-498, 2016.

[178] V. Gopalakrishnan, Y. Hu, and D. Rajan, "Random walks on graphs for salient object detection in images," *IEEE Trans. Image Process.*, vol. 19, no. 12, pp. 3232-3242, Dec. 2010.

[179] W. Wang, Y. Wang, Q. Huang, and W. Gao, "Measuring visual saliency by site entropy rate," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, San Francisco, CA, USA, Jun. 2010, pp. 2368-2375.

[180] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888-905, 2000.

[181] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, *et al.*, "Theano: A CPU and GPU math compiler in Python," in *Proc. Python Sci. Conf.*, Austin, TX, USA, Jun. 2010, pp. 1-7.

[182] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. Goodfellow, A. Bergeron, *et al.*, "Theano: new features and speed improvements," *arXiv preprint arXiv:1211.5590*, 2012.

[183] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998.

[184] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[185] M.-M. Cheng, N. J. Mitra, X. Huang, and S.-M. Hu, "Salientshape: Group saliency in image collections," *Visual Comput.*, vol. 30, no. 4, pp. 443-453, 2014.

[186] A. Vedaldi and K. Lenc, "MatConvNet-convolutional neural networks for MATLAB," *arXiv preprint arXiv:1412.4564*, 2014.

[187] R. Girshick, "Fast r-cnn," in *IEEE Int. Conf. Comput. Vision (ICCV)*, Santiago, Chile, Dec. 2015, pp. 1440-1448.

[188] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object Detection via Region-based Fully Convolutional Networks," *arXiv preprint arXiv:1605.06409*, 2016.

[189] N. Zhang, J. Donahue, R. Girshick, and T. Darrell, "Part-based R-CNNs for fine-grained category detection," in *European Conf. Comput. Vision (ECCV)*, Zürich, Switzerland, Sep. 2014, pp. 834-849.

[190] A. Rosenfeld and S. Ullman, "Visual concept recognition and localization via iterative introspection," *arXiv preprint arXiv:1603.04186*, 2016.

[191] N. Zhang, E. Shelhamer, Y. Gao, and T. Darrell, "Fine-grained pose prediction, normalization, and recognition," *arXiv preprint arXiv:1511.07063*, 2015.

[192] F. Zhou and Y. Lin, "Fine-grained image classification by exploring bipartite-graph labels," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 1124-1133.

[193] C. Li, X. Wang, Y. Xia, S. Eberl, Y. Yin, and D. D. Feng, "Automated PET-guided liver segmentation from low-contrast CT volumes using probabilistic atlas," *Comput. Methods Programs Biomedicine*, vol. 107, no. 2, pp. 164-174, 2012.

[194] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," *arXiv preprint arXiv:1511.05440*, 2015.

[195] T. Pfister, J. Charles, and A. Zisserman, "Flowing convnets for human pose estimation in videos," in *IEEE Int. Conf. Comput. Vision (ICCV)*, Santiago, Chile, Dec. 2015, pp. 1913-1921.

Appendix A

Abbreviations

AE	-	Auto-Encoder
BAOS	-	Boundary-Adjacent Object Saliency
BL	-	Bootstrap Learning based saliency detection
BMVC	-	British Machine Vision Conference
BS	-	Background Search
BSCA	-	Background Seed Cellular Automata based saliency detection
BSFE	-	Background Search and Foreground Estimation based saliency detection
CA	-	Context-Aware saliency detection
CB	-	Context-Based saliency detection
CNN	-	Convolutional Neural Network
CRF	-	Conditional Random Field
CVPR	-	IEEE International Conference on Computer Vision and Pattern Recognition
DC	-	Deep Convolution
DISC	-	Deep Image Saliency Computing
DL	-	Dense Labeling

DNN	-	Deep Neural Network
DRFI	-	Discriminative Regional Feature Integration based saliency detection
DSL	-	Dense and Sparse Labeling based saliency detection
DSR	-	Dense and Sparse Reconstruction based saliency detection
ECCV	-	European Conference on Computer Vision
FCN	-	Fully Convolutional Network
FE	-	Foreground Estimation
FES	-	Fast and Efficient Saliency detection
FT	-	Frequency-Tuned saliency detection
GC	-	Global Cues based saliency detection
GPU	-	Graphics Processing Unit
GR	-	Graph-Regularized saliency detection
GS	-	Geodesic Saliency detection
GT	-	Ground Truth
HC	-	Histogram-based Contrast
HDCT	-	High-Dimensional Color Transform based saliency detection
HS	-	Hierarchical Saliency detection
IA	-	Image Analysis
ICCV	-	IEEE International Conference on Computer Vision
IT	-	The saliency detection model proposed by Itti *et al.* in 1998
JOV	-	Journal of Vision
LEGS	-	Local Estimation and Global Search based saliency detection
LR	-	Low Rank matrix recovery based saliency detection

MAE	-	Mean Absolute Error
MC	-	Markov Chain based saliency detection
MCDL	-	Multi-Context Deep Learning based saliency detection
MDF	-	Multiscale Deep Features based saliency detection
MR	-	Manifold Ranking based saliency detection
MRF	-	Markov Random Field
PBO	-	Pseudo-Boolean Optimization based saliency detection
PCA	-	Principal Component Analysis based saliency detection
PDE	-	Partial Differential Equation
PR	-	Precision-Recall
RBD	-	Saliency optimization from Robust Background Detection
RC	-	Reversion Correction
ReLU	-	Rectified Linear Unit
RRWR	-	Regularized Random Walks Ranking
RW	-	Random Walks
SA	-	Saliency Aggregation
SAE	-	Stacked Auto-Encoder
SEG	-	SEGmenting salient objects from images and videos
SF	-	Saliency Filters for saliency detection
SL	-	Sparse Labeling
SLIC	-	Simple Linear Iterative Clustering
SM	-	Softmax
SPL	-	Signal Processing Letters

SR	-	Spectral Residual based saliency detection
SSD	-	Solid-State Drive
SUN	-	Saliency Using Natural statistics
SVM	-	Support Vector Machine
SVO	-	Salient Visual Objectness
TCSVT	-	IEEE Transactions on Circuits and Systems for Video Technology
TIP	-	IEEE Transactions on Image Processing
TNNLS	-	IEEE Transactions on Neural Network and Learning System
TPAMI	-	IEEE Transactions on Pattern Analysis and Machine Intelligence
UFO	-	Uniqueness, Focusness and Objectness based saliency detection