Delving into human visual attention for saliency detection of real-world images

Avishek Siris
789605

Submitted to Swansea University in fulfilment
of the requirements for the Degree of Doctor of Philosophy

Swansea University
Prifysgol Abertawe

Department of Computer Science
Swansea University

July 5, 2022

Declaration

This work has not been previously accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed .... (candidate)

Date 05/07/2022..................................

Statement 1

This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed .... (candidate)

Date 05/07/2022..................................

Statement 2

I hereby give my consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed .... (candidate)

Date 05/07/2022..................................

Abstract

Saliency detection explores the problem of identifying regions or objects that stand out from its surroundings. It is one of the fundamental problems in computer vision, with its application widely used in other graphics, vision and robotics tasks. Relative saliency ranking is a new problem that has been introduced with the idea of determining ranking based on the differences in the saliency agreement between multiple observers. This approach can lead to multiple objects being given the same saliency ranks. However, psychology studies and behavioural observations show that humans shift their attention from one location to another when viewing an image. This is due to the fact that the human visual system have limited capacity in simultaneously processing multiple visual inputs. We consider the sequential shifting of attention on objects as a form of saliency ranking, thus, we propose a new problem of saliency ranking based on attention shift. Although there are methods proposed for predicting saliency ranks, they are not able to model this human attention shift well. They are primarily based on ranking saliency values from binary prediction, which does not properly facilitate saliency rank reasoning between multiple individual objects. In this thesis, we aim to explore deep learning techniques for learning to rank salient objects by inferring human attention shift. We first construct a large-scale salient object ranking dataset. We define the saliency rank of objects by the order that an observer attends to these objects based on attention shift. We then propose a deep learning model that is built from bottom-up and top-down attention mechanisms for performing saliency ranking. Our model is evaluated with both quantitative and qualitative experiments, in which our proposed approach achieves state-of-the-art performance.

Regarding traditional salient object detection, we observe two main issues that lead to recent techniques failing in real-world complex image scenes. Firstly, most existing datasets consist of images with simple foregrounds and backgrounds, and limited number of objects that hardly represent real-life scenarios. Second, current methods only learn contextual features of salient objects with binary saliency labels. This is not very sufficient for a model to learn high-level semantics for saliency reasoning in complex scenes. We begin to address these problems

by constructing a new large-scale dataset with complex scenes rich in context. We then propose a context-aware saliency network that learns to explicitly exploit the semantic scene contexts of an image. We perform extensive experiments to demonstrate that our proposed network outperforms state-of-the-arts. The evaluation also show the effectiveness of leveraging high-level scene semantics for saliency detection in complex scenarios, while also transferring well to other existing datasets.

Acknowledgements

I would like to thank my supervisor Dr. Gary K.L. Tam, who has guided me throughout my studies in Master of Science and Doctor of Philosophy. Over the past years he has given both professional and personal advice, and valuable feedback and constructive criticism. He has always motivated me to do better and improve my skills in research, as well as personal and interpersonal skills. I will always be grateful for the time and effort he has spent to help succeed in my studies, and going the extra mile so I can achieve my best.

I would like to give special thanks to Dr. Jingjing Deng and Prof. Xianghua Xie for providing support throughout the years of my study.

I would also like to extend my gratitude to Dr. Rynson W.H. Lau, who was my supervisor outside of Swansea University. His guidance has also pushed me to reach for the top and improve the quality of my research. Additionally, I would like to thank Dr. Jianbo Jiao for his help in my research.

Finally, I would like to thank my family for their constant encouragement, support and believing in me to succeed over the course of my academic studies.

Contents

List of Publications **ix**

List of Acronyms **xi**

List of Tables **xii**

List of Figures **xiv**

1 Introduction **1**

1.1 Motivation 2

1.2 Contributions 4

1.3 Outline 5

2 Background **7**

2.1 Visual Attention 8

2.2 Saliency 9

2.3 Machine Learning 10

3 Related Work **15**

3.1 Salient Object Detection 16

3.2 Ranking in Saliency 19

3.3 Scene Context in Saliency 20

3.4 Attention Mechanisms 20

3.5 Saliency Datasets 22

4 Problem Statement **23**

4.1 Research Observations and Challenges 24

4.2 Research Hypothesis 25
4.3 Propsosed Ideas 26

5 Saliency Ranking 27
5.1 Introduction 28
5.2 Proposed Dataset 31
5.3 Proposed Method 37
5.4 Experiments 42
5.5 Limitations and Future Work 48
5.6 Summary 49

6 Saliency Ranking with Learning to Rank Supervision 51
6.1 Introduction 52
6.2 Proposed Method 54
6.3 Experiments 56
6.4 Limitations and Future Work 64
6.5 Summary 64

7 Scene Context Saliency 67
7.1 Introduction 68
7.2 Proposed Dataset 70
7.3 Proposed Method 75
7.4 Experiments 81
7.5 Limitations and Future Work 85
7.6 Summary 87

8 Combining Instance-based and Pixel-based Saliency 89
8.1 Introduction 90
8.2 Proposed Method 91
8.3 Experiments 94
8.4 Limitations and Future Work 97
8.5 Summary 98

9 Conclusions and Future Work 99
9.1 Conclusions 100

9.2 Future Work 101

Bibliography **105**

List of Publications

The following is a list of publications as a result from the works in this thesis.

Avishek Siris, Jianbo Jiao, Gary K.L. Tam, Xianghua Xie and Rynson W.H. Lau. “Inferring Attention Shift Ranks of Objects for Image Saliency”. Proceedings of the *IEEE/CVF Conference on Computer Vision and pattern Recognition (CVPR)*, 2020.

Avishek Siris, Jianbo Jiao, Gary K.L. Tam, Xianghua Xie and Rynson W.H. Lau. “Scene Context-Aware Salient Object Detection”. Proceedings of the *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

List of Acronyms

AI Artificial Intelligence

NN Neural Network

MLP Multi-layer Perceptron

CNN Convolutional Neural Network

FCN Fully Convolutional Network

ReLU Rectified Linear Unit

FPN Feature Pyramid Network

GT Ground-Truth

RNN Recurrent Neural Network

Conv Convolutional Layer

FC Fully Connected Layer

RCL Recurrent Convolutional Layer

Class Classification Layer

SOR Salient Object Ranking

MAE Mean Absolute Error

mAP Mean Average Precision

DCG Discounted Cumulative Gain

nDCG Normalized Discounted Cumulative Gain

GCN Graph Convolutional Network

List of Tables

5.1 Comparison with state-of-the-art methods on our dataset. Note that RSDNet scores are based on direct prediction with pre-trained weights from their dataset. ↑(↓) means the higher(lower) the better. Top two scores are shown in red and blue, respectively. 43

5.2 Ablation study of the proposed model. BbSR refers to the backbone network and the saliency rank network. 45

5.3 Quantitative comparison with S4Net for the task of salient instance detection on our dataset. Note that we do not include comparison with RSDNet, BASNet, CPD-R and SCRN since they are unable to perform this task. 47

6.1 Comparison with the state-of-the-art methods on our dataset (Section 5.2). The top section consists of salient object detection methods and middle section contains saliency ranking methods. The bottom section contains our methods, with ASSR [1] being our earlier model from Chapter 5. Note that RSDNet scores are based on direct prediction with pre-trained weights from their datasets. #Images used refers to the number of predicted images usable for SOR calculation. ↑ (↓) means the higher (lower) the better. Overall best performance is shown in **bold**. 58

6.2 Comparison with instance-based state-of-the-art methods on the dataset from [2] for testing generalisability. #Images used refers to the number of predicted images usable for SOR calculation. ↑ (↓) means the higher (lower) the better. Best scores are in red, while second best scores are in blue. 61

6.3 Quantitative comparison with S4Net for the salient instance segmentation task on our dataset. Note that we do not compare with other state-of-the-arts since they are unable to perform this task. 61

6.4 Ablation study of the proposed model. BbSR refers to the backbone network plus the small saliency rank network. 63

7.1 Comparison of the average number of objects and object categories per image among existing datasets and our dataset. 74

7.2 Quantitative comparison with state-of-the-art methods on our dataset. avgF refers to the average F-measure taken and E_m refers to E-measure. Red and blue indicate best and second best performances, respectively. 82

7.3 Quantitative comparison with state-of-the-art methods on five existing datasets to show *generalisability*. Note that we *train all comparison methods on our proposed dataset*, and *test directly* on these existing datasets for fair comparison. Furthermore, the test images for each dataset are filtered into a new subset that mainly include images that contain object categories defined in the proposed dataset. The number of images of the new subsets and the original test sets are respectively indicated next to the dataset. avgF refers to the average F-measure taken and E_m refers to E-measure. Red, Blue and Magenta respectively indicate the top 3 performance. 84

7.4 Ablation study of the proposed model on our dataset. Base: Mask-RCNN architecture, ISCS: Instance/Stuff Context Segmentation, SSCR: Semantic Scene Context Refinement, CIT: Contextual Instance Transformer. 85

8.1 Quantitative comparison with state-of-the-art methods on our dataset from Section 7.2. We also compare with our original model SCAS [3] from Chapter 7. avgF refers to the average F-measure, while maxF refers to the max F-measure. E_m refers to the E-measure and S_m to the S-measure. Red and blue indicate best and second best performances, respectively. 95

8.2 Ablation study of the proposed model on our dataset. Base: Mask-RCNN architecture, ISCS: Instance/Stuff Context Segmentation, SSCR: Semantic Scene Context Refinement, CIT: Contextual Instance Transformer, PsCSA: Pixel-based Saliency Stream with Contextual Saliency Aggregation 97

List of Figures

1.1 Two different types of saliency tasks. 2

2.1 Early visual attention model using bottom-up and top-down mechanisms for guided visual search [4]. 8

2.2 Mathematical process of a computational neuron in Neural Networks. 11

2.3 The LeNet convolutional neural network architecture introduced in [5]. 12

3.1 (a) architecture of early FCN-based saliency methods. (b) design of later saliency methods combining multi-level features through bottom-up and top-down pathways. (c) an iterative bottom-up and top-down saliency network proposed in [6]. 17

3.2 Each sub-figure (a-i) demonstrates various strategies for combining multi-scale features (coloured squares) in a Fully Convolutional Network [7]. 19

3.3 Relative ranking of salient objects (given by the grey-scale values) introduced in [8]. 19

3.4 The CapSal model from [9], which leverages images captioning to extract high-level scene semantics for salient object detection. Image Captioning Network (ICN) encodes the generated caption (CEV) to capture the semantic information of major objects. Local Perception Module (LPM) exploits bounding boxes to capture local context for segmenting salient regions in a local view, and utilizes the encoded caption feature to boost classification of bounding boxes. Global Perception Module (GPM) incorporates the caption feature with global visual features to localize salient objects in a more global view for a holistic estimation. The Fusion Module (FM) generates the final saliency map (S) by fusing the saliency maps produced by LPM (S_l) and GPM (S_g). 21

4.1 Example images from existing popular salient object datasets (e.g. , ECSSD [10], PASCAL-S [11], HKU-IS [12], DUT-OMRON [13] and DUTS [14]). Current saliency datasets are not very challenging and contain very few objects with simple background. Top row shows sample images from different datasets and the bottom row show corresponding ground-truth saliency. 25

5.1 Sample images from PASCAL-S dataset [11], which is used for saliency ranking in [8]. Note that multiple object can be given the same rank. The colours (orange→purple) indicate the saliency rank 1→5. 28

5.2 (a) Example input image, (b) corresponding ground-truth (GT) saliency rank, (c) corresponding GT saliency rank (colourised), (d) saliency rank prediction by RSDNet [8], (e) corresponding saliency rank by RSDNet (GT objects overlaid and colourised) and (f) corresponding saliency rank by RSDNet with only GT objects (overlaid and colourised). Note that the accuracy of predicting correct ranks can be impacted by the quality of the objects captured. 29

5.3 Sample images from our proposed dataset. Our dataset contains images with a maximum of 5 unique saliency ranks. 31

5.4 Screenshot of the annotation tool used by the participants during the user study. Participants are not told how the maps are generated. They are asked to pick a map that best respects the “order of attractiveness”. The green box indicates the map picked by one of the participants. 34

5.5 (a) Pick rates of maps by 11 participants in our user study across 2500 images. These maps are generated by nine methods that we experimented with in Section 5.2.1. (b) Distribution of ground-truth salient instances of all object categories in each data split of our dataset. 35

5.6 Average rank of each object category in the proposed dataset. 36

5.7 Architecture Overview. The model consists of a backbone network, Selective Attention Module (SAM), Spatial Mask Module (SMM) and a classification network for salient object ranking. We utilise Mask-RCNN [15] as our bottom-up backbone to provide object proposals with the FPN [16], and object segmentation from the segmentation branch. The bottom-up SMM extracts low-level features of the proposed objects while the top-down SAM considers high-level contextual attention features. 37

5.8 Details of the Selective Attention Module (SAM). Blue Block: object feature. Grey Block: pooled image feature. Green and Purple Blocks: fully connected layer applied onto corresponding features. Each block is a 512-D feature vector. . 39

5.9 Details of the Spatial Mask Module (SMM). Blue Block: 512-D object feature vector. Yellow Blocks: convolutional layers with output feature dimensions (96, 128, 64), kernel size (5×5, 5×5, 8×8) and strides (2, 2, 1). The three convolutional layers outputs a 64-D feature vector from an input 2D map (32×32) that represents the spatial information for each object. 41

5.10 Comparison of the proposed method with state-of-the-art methods: RSDNet [8], S4Net [17], BASNet [18], CPD-R [19] and SCRN [20]. Each example in the top row shows the input image, ground-truth saliency map and ground-truth ranks, while for the following rows: (i) saliency prediction map, (ii) saliency prediction map with predicted rank of ground-truth object segments colourised on top, and (iii) corresponding map that contains only the predicted rank of ground-truth objects. The result in (iii) is leveraged to obtain the predicted saliency ranks for quantitative evaluation. 46

5.11 Example scenes containing "sports ball" object category. Images from our dataset (Top row), GT Ranks (Middle row), our network rank prediction (Last row). 48

6.1 (a) Example input image, (b) corresponding ground-truth (GT) saliency rank, (c) corresponding GT saliency rank (colourised), (d) saliency rank prediction by RSDNet [8], (e) corresponding saliency rank by RSDNet (GT objects overlaid and colourised) and (f) corresponding saliency rank by RSDNet with only GT objects (overlaid and colourised). Note that the accuracy of predicting correct ranks can be impacted by the quality of the objects captured. 53

6.2 Architecture Overview. In this work, we modify our previous saliency rank network from Chapter 5. Specifically, we enable end-to-end training, adjust network parameters, introduce a new Salient Instance Edge Module (SIEM) and transition from rank-id classification to rank score prediction. 54

6.3 Details of the Salient Instance Edge Module (SIEM) and Mask Segmentation. Orange Blocks: mask segmentation features. Green Blocks: edge features. The orange and green blocks represent features obtained after applying convolutional layers. Both mask segmentation and edge features are of shape (14×14) with 256-D feature channels. Transposed convolutional layers (5th blocks) increases the mask segmentation and edge features to size (28×28). The final orange and green blocks (with lighter shade) represent the mask segmentation and edge predictions for individual objects. 56

6.4 Comparison of the proposed method against state-of-the-art methods designed for saliency ranking: ASSR [1], SOR-PPA [21], IL-RSR [2] and RSDNet [8]. Each example in the first row shows the input image, while the second row show ground-truth saliency map (i) and ground-truth saliency ranks (ii). The following rows are predictions from the compared methods, where (i) saliency prediction map, (ii) corresponding map that contains only the predicted rank of ground-truth objects. The result in (ii) is leveraged to obtain the predicted saliency ranks for quantitative evaluation. 60

6.5 Simple illustration of the GT saliency rank generation process between the dataset in [2] and ours (Section 5.2). [2] uses the max intensity value inside the segmentation of objects from the fixation map to assign saliency rank in descending order (left). Our GT rank order is based on unique attention shifts between objects using the sequential fixation data (right). 60

6.6 Qualitative comparison of instance-based salient object segmentation. Instances with 5 different shades of blue are predicted instances that match with their corresponding GT instance. Red instances represent false predictions. 62

6.7 Qualitative comparison of salient object segmentation between our full model with Salient Instance Edge Module (SIEM) versus full model without SIEM. 63

6.8 Example scenes containing objects in the *vehicle* category. Input images from our dataset (Top row), GT Ranks (Middle row), rank prediction from our method (Last row). 64

7.1 Examples of real-world complex scenarios where existing methods (e.g. , CPD-R [19], EGNet [22], SCRN [20], LDF [23] and MINet [24]) may not capture semantic scene contexts well, leading to incorrect detection of distractors. 68

7.2 Sample images (Top row) and corresponding ground-truth saliency (Bottom row) from the CapSal datatset [9] and our proposed dataset. CapSal contains complex images, but the ground-truth saliency is heavily biased towards the captions data. Whereas, our ground-truth saliency is entirely based on fixation data. 70

7.3 Example images of overlapping segmentations that often cause issues in the automatic ground-truth saliency generation. (a) image, (b) foreground object, (c) background object, (d) resulting incorrect saliency map. Red cross in (b) and (c) correspond to an observer's fixation. 72

7.4 Examples of certain object categories (e.g. , car and train) correctly and incorrectly identified as salient in separate scenes, with different object sizes. (a) and (b) are image and corresponding generated saliency map, with those object categories correctly defined as salient, during phase 1 of automatic saliency generation. (c) and (d) are image and resulting saliency map generated with incorrect saliency. The same object categories (car and train) are relatively large and should be considered as background instead. 72

7.5 Examples of visual comparison during the manual image filtering (Phase – 2) when we construct the proposed dataset. In (a) and (b), the generated saliency maps are comparable to the corresponding SALICON fixation maps, which are kept in our final dataset. Whereas those images with larger discrepancies, as shown in (c) and (d), are removed from the dataset. 73

7.6 Statistics of the proposed dataset, presenting the distribution of all objects (*Things*) and *Stuff* region categories in the dataset. 74

7.7 Further statistics of the proposed dataset. (a) and (b) reports the distribution of size and distance from image centre of salient objects, respectively. (c) average number of all objects (Things), salient objects and regions (Stuff). (d) shows an intensity map from overlays of all individual objects (i), salient objects (ii) and Stuff regions (iii). Local (e) and global (f) colour contrasts of salient objects. 75

7.8 An overview of the proposed network. Our model extracts semantic features from the Shared Context Segmentation Decoder. The decoder is trained to reconstruct features for generating *Things* and *Stuff* categories. Our Semantic Scene Context Refinement (SSCR) module then utilizes the semantic features and multi-scale features to build the augmented scene context features, correlating the semantics of an image. Our Contextual Instance Transformer (CIT) module inside the Salient Instance Network, learns relationships between objects and scene context, and enhance saliency reasoning. 76

7.9 Details of the Semantic Scene Context Refinement (SSCR) module. 77

7.10 Details of the Contextual Instance Transformer (CIT) module. 80

7.11 Details of the Salient Instance Network. 81

7.12 Qualitative comparison of the proposed method with ten state-of-the-art saliency methods: BASNet [18], CapSal [9], CPD-R [19], PFANet [25], S4Net [17], EGNet [22], SCRN [20], ITSD [26], LDF [24] and MINet [23]. 83

7.13 Example images in common datasets (ECSSD [10], PASCAL-S [11], HKU-IS [12], DUT-OMRON [13] and DUTS [14]) that contain simple scenes. There are only one or very few objects. The object categories are not defined in MS-COCO [27] and the datasets do not contain semantic segmentation annotations. As a result, our network is unable to train on these datasets and direct testing does not always lead to our network achieving new state-of-the-art performances (Table 7.3). Images (Top) and corresponding ground-truth saliency (Bottom). 85

7.14 Visual comparison of our proposed method with state-of-the-arts (BASNet [18], CPD-R [19], PFANet [25], S4Net [17], EGNet [22], SCRN [20], ITSD [26], LDF [24] and MINet [23]) on existing salient object detection datasets (ECSSD [10], PASCAL-S [11], HKU-IS [12], DUT-OMRON [13] and DUTS [14]). 86

8.1 Architecture Overview. We extend our salient object detection method from Chapter 7. We introduce a pixel-based saliency stream and combine its prediction with our original instance-based saliency. The combination is performed by the Contexual Saliency Agggregation (CSA) module, which learns to dynamically fuse saliency predictions from the two streams. 91

8.2 Details of the Pixel-based Saliency Stream. 92

8.3 Details of the Contextual Saliency Aggregation (CSA) module. 93

8.4 Qualitative comparison of the proposed method with state-of-the-art saliency methods: BASNet [18], CapSal [9], CPD-R [19], PFANet [25], S4Net [17], EGNet [22], SCRN [20], ITSD [26], LDF [24], MINet [23], CSNet [28], GateNet [29] and VST [30]. We provide further comparison with our original model, SCAS [3] (from Chapter 7). 96

Chapter 1

Introduction

Contents

1.1 Motivation **2**

1.2 Contributions **4**

1.3 Outline **5**

1.1 Motivation

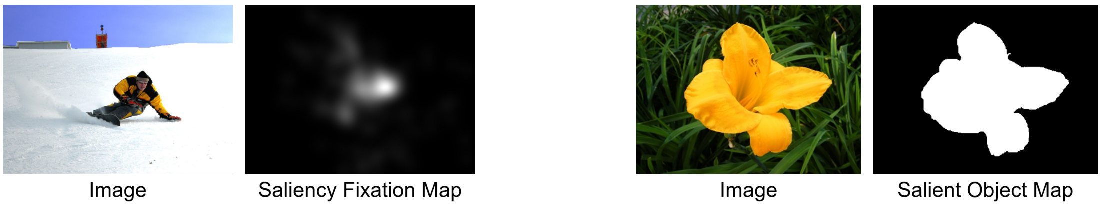


Figure 1.1: Two different types of saliency tasks.

Saliency is a concept that describe parts of an image scene, which are considered to attract the visual attention of a human observer. Those highlighted parts are either regions or objects that are seen to stand out from their surroundings [31]. The saliency of an image is usually defined based on the capture of eye-tracking data. Saliency has been modelled in two ways (shown in Figure 1.1), the first involves in generating a heat-map like saliency map that highlight areas considered to be salient. The second is formulated as a task of salient object detection, which the modelling process consists of finding and segmenting salient objects only [32]. In this work we will focus on segmenting salient objects.

The task of salient object detection has seen great research interest in areas from computer vision, graphics and robotics [31]. One of the contributing factors is that it helps in understanding and modelling the mechanisms of human visual attention. It is also widely used in many computer vision tasks as a pre-processing step, which includes image captioning [33], image compression [34], image parsing [35], person re-identification [36, 37] and video object segmentation [38]. For instance, [33] use an attention mechanism to help an image caption generation model to focus on different relevant image locations for each word generation. This also provides the insight and visualization of “where” and “what” is focused when generating the image caption. Their experiments show that the exploitation of attention improves the interpretability of the image caption generation and aligns well with human intuition. However, there is always a potential issue if saliency detection is not predicted well or does not align well with a task, then it can mislead further processing when relied upon in downstream applications. It is one of the many reasons why research in saliency detection strive to improve the generalizability of saliency prediction in diverse scenarios.

Many studies in saliency modelling has been performed separately with bottom-up or top-down visual attention. Bottom-up visual attention is stimulus driven, whereas top-down attention is voluntary and task-driven, requiring prior knowledge [31]. For example, bottom-up attention can be driven by the contrast of colour or shape of an object compared to its surrounding. While top-down attention may be driven by a task such as driving, where the attention of the driver can be influenced by road signs and actions required to react with the situations on the road. Although studies have shown some success, modelling attention separately do not follow closely to actual human visual mechanism of processing bottom-up attention first, followed by top-down. Recent saliency methods utilize the power of deep learning techniques to improve the accuracy of salient object detection. These methods are mainly based on Convolutional Neural Networks (CNN) [39], which are used to learn saliency features from large datasets. In Chapter 3, we provide a survey of bottom-up and and top-down methods for saliency detection, as well as deep learning techniques.

From our review of the saliency literature, we observe potential limitations with current saliency techniques: (1) explicitly combining bottom-up and top-down attention for saliency is not fully explored. Using bottom-up low-level features (e.g. , contrasts in colour, intensity, orientation, texture) alone is insufficient and it would benefit to combine both bottom-up and top-down attention together, in order to predict visual attention with high accuracy [40]. (2) despite the fact that recent saliency methods have achieved significant performance, these methods still have difficulties when predicting on images with complex scenes. (3) many methods [18, 19, 22–25] have been proposed to model salient object detection as a binary prediction problem, where all predicted objects are given the same value and importance. Humans, however, are shown to have the ability to sequentially select and shift attention from one region/object to another [41,42]. Modelling this ability is important for the understanding of how humans interpret images, and helps improve the performance of relevant applications (e.g. , autonomous driving [43] and robot-human interactions [44]). Here, we interpret and define this human ability as *Saliency Ranking*.

The objective of this work is to investigate the combination of bottom-up and top-down attention for saliency detection. We look into two tasks, namely salient object ranking and traditional salient object detection. In both tasks, we first tackle the issues with current datasets in regards to the corresponding tasks, which include the focus of saliency detection on complex image scenes. We then compare existing state-of-the-art methods and present a novel deep-learning framework for both tasks. We further extend our work by building upon the issues and limitations of our approaches and state-of-the-arts.

1.2 Contributions

During our investigation of saliency techniques, we discover that human attention shifts can be modelled by deep learning methods, which is supported by our experimental results. Moreover, we can improve saliency detection with the combination of bottom-up and top-down attention, while also performing on complex image scenes. The main contributions of this work can be seen as follows:

- **New research problem of salient object ranking (Chapter 5)**

  We propose a new research problem to predict salient object ranks according to human attention shift. It is inspired by psychological and behavioural studies. It goes beyond just human-object interactions, and can also model object-object attention shift

- **New large-scale dataset for salient object ranking (Chapter 5)**

  We propose a new large-scale dataset for the problem of salient object ranking based on attention shift, justified by our user study.

- **Deep-learning method for salient object ranking (Chapters 5 & 6)**

  We propose a deep learning approach to jointly predict saliency ranks of multiple salient object instances and their corresponding object masks, with bottom-up and top-down attention mechanisms.

- **New large-scale dataset for salient object detection (Chapter 7)**

  We build a new salient object detection dataset with real-world complex scenes to consider semantic scene contexts.

- **Deep-learning method for salient object detection (Chapters 7 & 8)**

  We propose a semantic scene context-aware framework for salient object detection, which explores the semantic relationship between salient objects and the scene context. The framework is specifically designed to target salient object detection in complex image scenarios.

- **Comprehensive comparison of deep-learning saliency methods (Chapters 3-8)**

  We showcase comprehensive comparison of current state-of-the-art deep-learning methods for both tasks of salient object ranking and salient object detection. The comparison is run as fairly as possible using the available source code from published works. The aim of this comparison is to examine the performance between current state-of-the-arts and our proposed methods.

- **Data and source code (Chapters 5 & 7)**

  We release our proposed datasets and methods defined in this work for further research and public use.

1.3 Outline

The remaining chapters of this thesis is structured as: **Chapter 2** provides an introductory background of the related concepts in this work. **Chapter 3** reviews related work corresponding to salient object detection, ranking in saliency, scene context in saliency and attention mechanisms. **Chapter 4** discusses the problems we have observed from existing works and present our research hypotheses based on the challenges and limitations we have discovered. We then propose research ideas to explore the observed problems. In **Chapter 5**, we justify with as a user study, a method for generating ground-truth salient object rank data. We then design a novel deep-learning approach for salient object ranking, based on bottom-up and top-down attention mechanisms. Experimental results are shown to demonstrate state-of-the-art performance. **Chapter 6** extends the work of **Chapter 5**. We improve upon the limitations of the proposed deep-learning approach. Further experiments are performed showcasing new state-of-the-art performance. In **Chapter 7**, we build a new large-scale salient object detection dataset targeting complex real-world image scenes. We then detail a novel framework that leverages semantic scene context information for salient object detection in complex scenarios. Experimental results show improvements on state-of-the-art methods. **Chapter 8** also extends the work of **Chapter 7**. We enhance our salient object detection framework, by tackling its issues and limitations. Additional experiments showcase the extended work producing new state-of-the-art results. In **Chapter 9**, we draw our final thoughts and conclude the work presented in this thesis. We reiterate the contributions of this work and consider potential ideas for future research.

Chapter 2

Background

Contents

2.1 Visual Attention **8**

2.2 Saliency **9**

2.3 Machine Learning **10**

2.3.1 Neural Networks 11

2.3.2 Convolutional Neural Networks 12

2.3.3 Fully Convolutional Networks 13

In this chapter, we provide a background overview of the concepts relating to salient object detection. We first present the current understanding of human visual attention (2.1). Then, we discuss the ideas of saliency (2.2), followed by machine learning techniques (2.3). We would like to state that the background mentioned in this chapter, touches upon the concepts related to this work as a reminder to understand the following chapters. We refer readers to the corresponding references in each subsection for further information.

2.1 Visual Attention

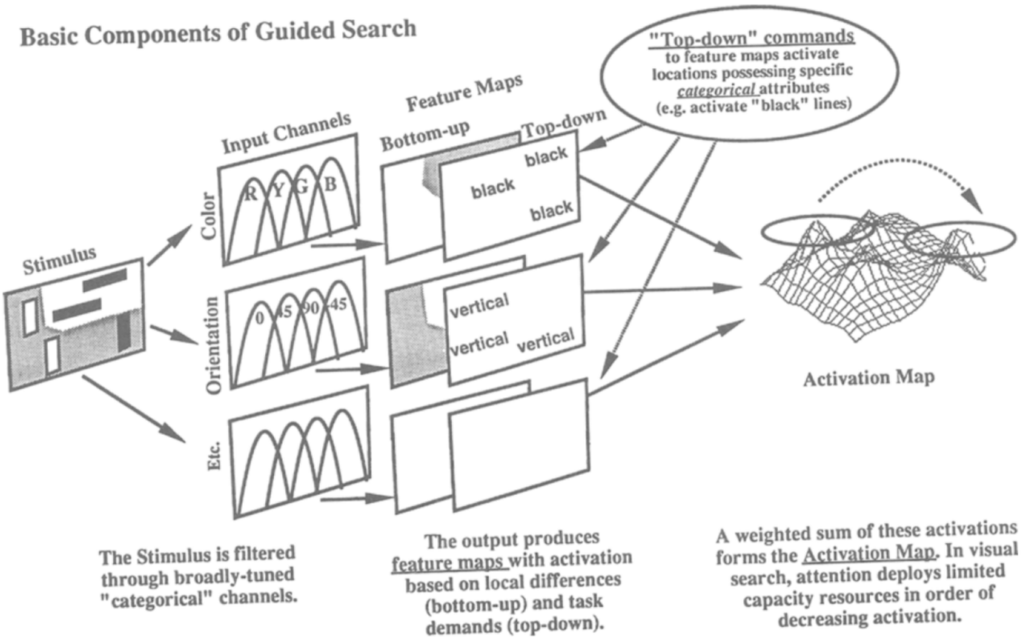


Figure 2.1: Early visual attention model using bottom-up and top-down mechanisms for guided visual search [4].

Human sight is facilitated by the retina which produces high-resolution central fovea with low-resolution periphery [31]. The human visual system has limited capacity for processing the large amounts of visual information that is perceived from sensory stimuli. Given the limited resources, humans have evolved to process multiple simultaneous visual inputs through mechanisms of selective visual attention [45]. This process involves in sequentially selecting and shifting attention from one region/object to another [41, 42].

Numerous psychology [46] and neuropsychology [47, 48] works have investigated the behaviour and functions of visual attention. Visual attention is generally categorized into two mechanisms, bottom-up attention and top-down attention. Bottom-up attention is purely driven by low-level stimuli that are considered to pop out from their surrounding background. Top-down attention is usually referred to as guidance from high-level information such as prior knowledge, and task-driven goals [48]. Both bottom-up and top-down attention are usually described as distinct mechanisms that influence human attention. However, the two attention mechanisms actually interact and influence one another for orienting visual attention. Early models of visual search [4] also demonstrate that attention is captured by simultaneous integration of bottom-up and top-down mechanisms. Visual attention modelling is performed by computing a fixation map that gives a probability of where a viewer's attention/gaze will likely be attracted towards. Ground-truth (GT) fixation data is generated by capturing eye-tracking data from a viewer observing an image with an eye-tracking device. The ground-truth fixation data is then used to compare and evaluate the performance of visual attention models.

2.2 Saliency

Saliency is a term that is frequently used interchangeably with visual attention and gaze. Similarly, it describes a region or object that stands out and grab visual attention of an observer. Saliency maps, like-wise to fixation maps, are used to model the saliency of regions and their strength of attracting attention based on the intensity of their saliency values.

Early saliency work implemented handcrafted and computational models that were mainly based on bottom-up attention mechanisms [42,49]. These models relied on the combination of low-level features (e.g. , colour, intensity, orientation) contrast, in order to generate the saliency map.

Current research trend has emerged from fixation prediction to salient object detection, after the seminal work in [50] that instigated a new research direction of segmenting salient objects. Salient object detection involves in producing a saliency map that segment objects that are considered to be salient. The task is similar to the fundamental segmentation problem [39] in computer vision that involves segmentation of objects, but with the focus on only segmenting foreground salient objects. In general, saliency object detection requires for a model to correctly identify true salient objects, while suppressing false background objects. Secondly, it demands for the segmentation quality of salient objects to be highly accurate. Saliency models are also desired to be computationally efficient, as the models are widely used for pre-processing in further applications. Recently, saliency object detection has progressed from hand-crafted methods to the adoption of machine learning techniques, due to the rising popularity of Convolutional Neural Networks (CNN) [51, 52]. CNN-based methods do not rely on low-level contrast and removes the need for biases, which are often employed in general computational methods. For instance, center bias is a commonly used prior that assumes a salient object is likely to be found near the image center [7]. It relates to the fact that in most images, human photographers tend to place objects of interest in the center of photographs, and experimental setup for capturing eye fixation data usually place human participants centrally in front of a screen [53]. Furthermore, psychology studies [54] further demonstrate that observers have a strong tendency to attend around the center of a scene. However, priors like the center bias may not always be useful in certain images (e.g. , scene with no objects at center), but they do provide a good statistical cue for guiding saliency and suppressing background based on the characteristics of a dataset. CNN-based methods are also able to train on large-scale datasets, with great quantity of tunable parameters that can effectively learn to capture various saliency features. The significant performance gain and desirable properties of CNN-based methods has made them a popular approach for saliency modelling.

2.3 Machine Learning

Artificial Intelligence (AI) is the science of understanding and building intelligent entities. It has been defined as the study of agents that receive information from its environment and perform corresponding actions [55]. A rational agent is characterized as an agent that behave and perform actions that achieves the best possible outcome, which can be captured by some performance measure based on a task or the environment.

Machine learning introduces a learning paradigm in artificial intelligence. It can be categorized as supervised or unsupervised learning. Supervised learning takes in input and target output samples, then learns a function that maps from input to output, mainly through an iterative process until a criteria is met that well satisfies some objective function [56]. Unsupervised learning is based on the learning of patterns and probability distributions of datasets without the use of target labelled data [57]. There are other machine learning techniques, including semi-supervised, zero-shot/one-shot and self-supervised learning that also use zero to few amounts of labelled data. These techniques are more challenging as the training procedure is not as simple like supervised training and the output is unknown during training in most cases. We will not go into further detail here and we will focus on supervised learning in this work, as current supervised learning methods achieve the best performance for saliency detection.

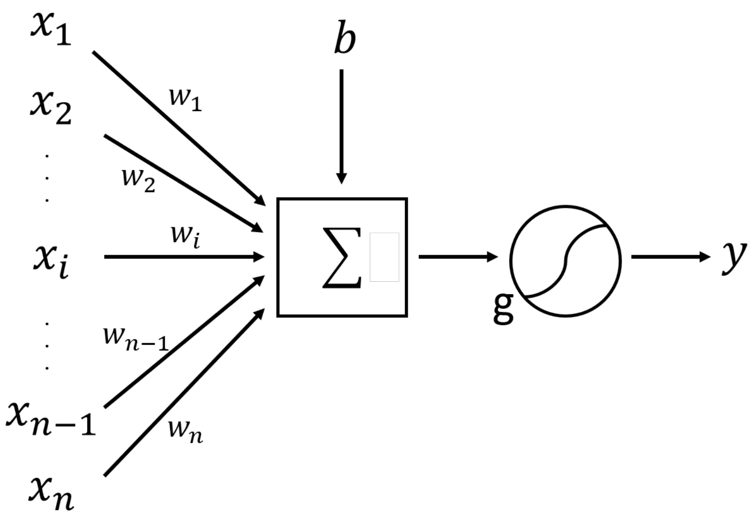


Figure 2.2: Mathematical process of a computational neuron in Neural Networks.

2.3.1 Neural Networks

A Neural Network (NN) is an algorithm that was inspired from biological neural activity [58]. Neural networks are composed of nodes that are analogous to neurons. It is designed to compute a weighted sum of all its inputs, plus a bias before applying an activation function (e.g. , sigmoid) to calculate the output [56]. Figure 2.2 illustrates this mathematical process. The process is thought to correlate to the idea that when a combination of inputs exceed some threshold, a signal is fired by the activation function. Given inputs x, we compute the output y as:

$$y = g(b + \sum_{i}^{n} x_i w_i), \tag{2.1}$$

where, x_i is an input connection from node i and w_i is the corresponding learnable weight for that connection. b is the bias added to the summation and g is the activation function applied for the final output. A network with a single node is called a Single-Layer Perceptron, while connecting multiple nodes in stacks and one after the other builds a Multi-Layer Perceptron (MLP). MLP allows mapping of more complex and robust functions, which a single threshold function would not be able to handle. It further emulates the massive number of neurons linked together in the human brain, where signals are fired based on the stimulation of multiple inputs.

Neural Networks are trained by updating its learnable weights by a factor determined by the error of its output. The error is calculated by some loss function that is based on the task in hand. For example, the cross-entropy loss is a common loss function that is used to calculate the error in classification tasks. Gradient descent optimization is then used to pass back the error though the network to update the weights [56]. With the addition of hidden layers (layers between the input and output layers) in MLPs, it introduces difficulty in calculating the error for those hidden layers, as the inputs of a layer is based on the output of the previous layer. This is where the back-propagation algorithm comes into effect. The idea behind the algorithm is to consider that each node in the hidden layer is responsible for a fraction of the error and so, its weight must be updated accordingly [56]. More specifically, a partial derivative is calculated for each node with respect to its inputs and weights. The partial derivative is then added to the weight of the corresponding node for update:

$$w_{i,j} = w_{i,j} + \Delta w_{i,j}, \tag{2.2}$$

$$\Delta w_{i,j} = \frac{\partial \varepsilon}{\partial w_{i,j}}, \tag{2.3}$$

where, ε is the error and $w_{i,j}$ is the weight for the connection between nodes i and j from adjacent layers.

2.3.2 Convolutional Neural Networks

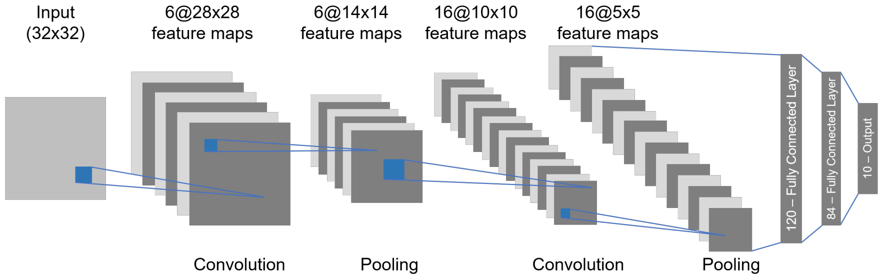


Figure 2.3: The LeNet convolutional neural network architecture introduced in [5].

Convolutional Neural Networks (CNNs) was first introduced in [5], which has now become the most common method for machine learning. It solved the issues of MLP in computer vision tasks, which included the poor scalability of training many hidden layers, too many parameters required to train and non-localised features. CNNs employ the convolution operation with small filters. The filters are applied across the image, allowing the filters to detect certain types of local features from different locations in an image. This further helps reduce the number of parameters for training, as the weights are shared for a given filter. Consequently, it enables better scaling with additional hidden layers. The early shallow layers in CNNs are found to model abstract and low-level features such as the edges of an object [59]. As you progress from mid to deeper layers, the features encoded in those layers begin to take shape and model structural parts of an object.

Pooling layers are usually applied after every few convolutional layers. It reduces the resolution of the output features and generalizes the high amount of spatial information. A pooling layer with a (2x2) pooling kernel is usually applied. As the network gets deeper, the spatial resolution is typically halved, while the number of convolution filters is increased. Max and average pooling are the common pooling techniques used in CNNs.

The first fully connected layer in a network usually flattens features (output) from the previous layer into a 1-D feature vector. It then multiplies the feature with a weight matrix. Fully connected layers are usually added at the end of CNNs to output some probability pertaining to a task for a given input image (e.g. , probability that an image contains a cat).

2.3.3 Fully Convolutional Networks

Fully Convolutional Networks (FCNs) are networks that are mainly built from convolutional layers and do not contain any fully connected layers. One advantage of FCNs over CNNs is that they can take any arbitrary sized input and produce a correspondingly sized output [39]. FCNs are also able to retain spatial information as there are no fully connected layers which requires features to be reduced into 1-D vectors. Another benefit is that FCNs are more efficient and requires less resources (memory and computation power). This comes from the fact that convolutional layers are relatively less resource intensive, as it shares weights and requires less number of connections. Whereas, fully connected layers connect every neuron from the current layer to every neuron in the previous layer, with each connection having its own weight.

Recent salient object detection methods are built from FCNs and deep learning techniques. Deep learning is a subset of machine leaning, which generally consists of large complex networks with very deep layers. Deep learning networks are usually trained with tremendous amounts of data, as they are able to scale very well with increasing data. With the advancements in computational hardware and the availability of large-scale labelled datasets, current trends adopt deep learning techniques to improve performance, while also facilitating the progression of research towards more complex vision tasks.

Chapter 3

Related Work

Contents

3.1 Salient Object Detection **16**

3.1.1 Bottom-up and Top-down Approaches 16

3.1.2 Deep-learning Based Methods 18

3.2 Ranking in Saliency **19**

3.3 Scene Context in Saliency **20**

3.4 Attention Mechanisms **20**

3.5 Saliency Datasets **22**

In this chapter, we begin by reviewing traditional salient object detection. We divide this first section into bottom-up and top-down approaches (3.1.1) and deep-learning methods (3.1.2). Then, we examine ranking in saliency (3.2) and scene context in saliency (3.3), which specifically target the tasks of salient object ranking and salient object detection in this work. Finally, we analyse attention mechanisms used in saliency (3.4).

3.1 Salient Object Detection

3.1.1 Bottom-up and Top-down Approaches

Salient object detection can also be categorised into bottom-up, top-down, or a combination of both. Here, we focus on those that combine both bottom-up and top-down approaches. Early methods that combine bottom-up and top-down approaches use hand-crafted and computational based features. Bottom-up features usually come from local and global contrasts in color, intensity and orientation [53]. Top-down features often relate to the specific tasks at hand and generally encode some prior knowledge that can guide attention. Notable examples of top-down features include using high-level face features [60], photography bias [53], and person and car detector [40]. Photography bias is similar to center bias where objects of interest are usually placed at the center of a photo. Humans and faces humans are found to naturally attract the attention of human observers, thus, they can be used as additional bias to guide saliency. Gist features [61] can provide a statistical cue to where salient obejcts may likely be located, and gaze patterns learnt from performing specific tasks [62] can influence the attention of an observer.

With the advance of Convolutional Neural Networks (CNNs), CNN features are leveraged to improve the performance of saliency detection [28]. Some works [63] use a simple stack of convolution and deconvolution layers, while some others [12, 64] design multi-scale networks to capture contextual information for saliency inference. Recent studies further incorporate a top-down pathway [65–67]. High-level semantics in the top-layers are refined with the low-level features in the shallow-layers through side connections [24, 25, 29, 68]. The refinement generates a better representation at each layer [69] and is thought to imitate the bottom-up (low-level stimuli) and top-down (visual understanding) human visual process [6], shown in Figure 3.1.

Wang *et al.* [70] follow the relationship between eye fixation and object saliency previously studied in [11] and propose to use fixation maps to guide saliency in a top-down manner.

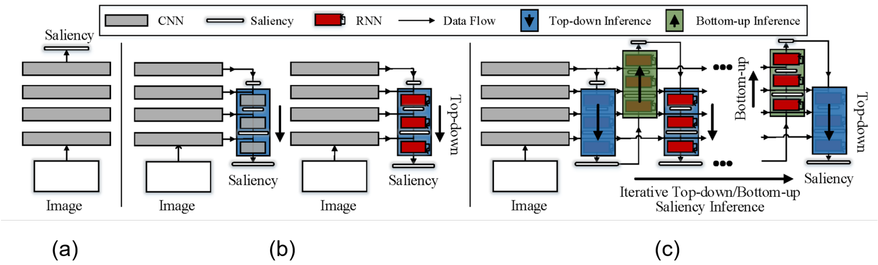


Figure 3.1: (a) architecture of early FCN-based saliency methods. (b) design of later saliency methods combining multi-level features through bottom-up and top-down pathways. (c) an iterative bottom-up and top-down saliency network proposed in [6].

Recently, Zeng *et al.* [71] proposed to unify the task of salient object detection and weakly-supervised semantic segmentation. They introduced a saliency aggregation module that used saliency scores to weight corresponding semantic segmentation, in order to generate the final saliency map. Li *et al.* [72] pair salient object and camouflaged object detection to learn contradictory information and enhance performance of the two tasks. Aydemir *et al.* [73] used object detection to produce dissimilarity scores based on visual appearance and relative size, so as to enhance the saliency contrast among objects.

The above methods mimic the human visual process using both bottom-up and top-down pathways. However, these methods mainly approach bottom-up attention by either obtaining low-level contrast features or only considering shallow layers in a CNN. This can somewhat limit the representation of bottom-up attention to a set of specific features only. Moreover, low-level contrast is unlikely to be very effective in complex real-world image scenes, where there is often low contrast, or very cluttered features and distractors (background regions/objects with similar visual features to salient objects). It would be refreshing to explore alternative approaches to obtain bottom-up attention besides low-level colour and texture contrast, while relieving the difficulties of low-contrast scenes. In terms of top-down attention, existing methods mostly use bias, detection of people or regard deep-layers in CNNs for top-down attention features. There is many more possibilities in generating representation for top-down attention as they could be extracted from auxiliary tasks or prior information, and it would be interesting to investigate various approaches.

3.1.2 Deep-learning Based Methods

Previous deep learning salient object detection methods used multi-layer perception (MLP) to predict a saliency score for each pixel in an image [12, 74–76]. Though these MLP-based models outperform traditional hand-crafted saliency methods [77, 78], they were unable to capture spatial information effectively due to the use of fixed fully connected layers. Later methods tackled this issue by utilizing fully convolutional networks (FCNs) [39], building their success on semantic segmentation.

Many of the recent deep learning based saliency models were built on the FCNs with various strategies to combine multi-scale contextual information. They mostly embedded modules for extracting and aggregating context features from different layers in the network [20, 24, 28, 29, 67, 79]. Typically, they employed side outputs from different layers in their encoders, and aggregated those side outputs with the layers in their decoders [6, 22, 23, 26, 65, 69, 70, 80]. Su *et al.* [81] further extracted multi-scale contextual features using varying dilated convolutions, while recurrent blocks were applied in [51] and [82]. Subsequent works proposed to explicitly combine local and global contextual features through (a) the use of separate networks [83, 84], (b) additional convolutions after the final convolutional layer in an encoder [85, 86] and (c) adopting a Pyramid Pooling Module [87, 88].

In [22] and [26], saliency features are complemented with edge information at various resolutions, in order to improve the accuracy of salient object segmentation and their boundaries. Wei *et al.* [23] further propose to decompose saliency maps into body and detail maps. The body map contains the central area of objects, while the detail map focuses on the boundaries of objects.

Although the above-mentioned saliency methods have shown significant improvements, they still struggle with complex scene images that are rich in semantic context. A contributor to this issue comes from the fact that existing methods are usually designed to train on datasets, which contain simple images with very few objects and background distractors. Additionally, these methods mainly learn a limited scope of discriminative spatial context features in multiple scales. The networks are generally trained with binary saliency annotations only, and fail to effectively learn high-level semantics. It would be interesting to incorporate high-level semantic information from complementary tasks in order to learn more discriminate context features, while targeting complex image scenes with rich semantics.

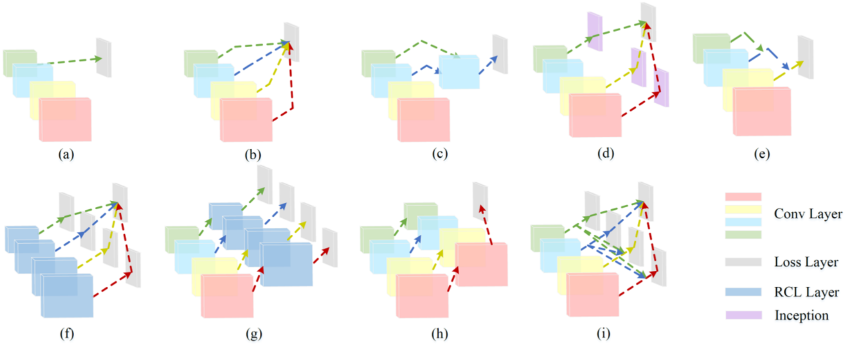


Figure 3.2: Each sub-figure (a-i) demonstrates various strategies for combining multi-scale features (coloured squares) in a Fully Convolutional Network [7].

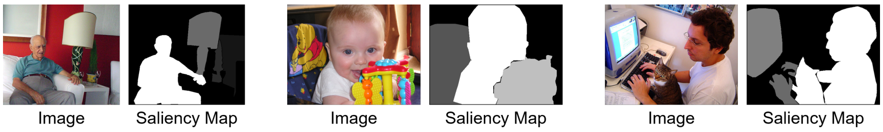


Figure 3.3: Relative ranking of salient objects (given by the grey-scale values) introduced in [8].

3.2 Ranking in Saliency

Ranking of salient objects is a relatively new problem that attempts to segment and rank order individual salient objects (more details discussed in Chapter 4). It is introduced by Islam *et al.* [8], in which they define object ranks as the *degree of agreement* among multiple observers who consider if objects are salient. The extended work of [8] in [89] use the same patch-based network architecture, while introducing a new dataset for saliency ranking. A network proposed in [2] follows closely to [8, 89], and employs a graph-based module for learning to rank with a pair-wise ranking approach. [8, 89] provides a good introduction to saliency ranking, however, their definition of saliency ranking is based on the relative difference in saliency agreement between multiple observers. Additionally, parts of their ground-truth data contain multiple objects with tied saliency rank in the same image. Both the definition and the ground-truth data do not match well with past psychological studies and behavioural observations [45], where multiple attentions of foci is not supported [47] and attention is sequentially shifted. Performing saliency ranking that is closer to the human visual attention system would be a good direction for research, as it would lead us to develop models that can more closely simulate human attention behaviour.

Fang *et al.* [21] is a recent work that follows our study (discussed in Chapter 5). They propose a network with similar backbone to ours and a Position-Preserved Attention module to incorporate positional information with object features for salient object ranking.

In the literature, there are other works that use ranking techniques for saliency estimation. For example, [90] use graph-based manifold ranking for saliency inference. [91] also incorporates rank learning to select visual features that best distinguish salient targets from real distractors. However, these works use ranking as a formulation to output a final binary saliency prediction and do not predict saliency rank order of multiple objects.

3.3 Scene Context in Saliency

Gist features are considered as an abstract low-level scene representation. Torralba *et al.* [92] combined scene representation from holistic low-dimensional encoding with low-level saliency in a statistical framework for modelling attention. Peter *et al.* [61] proposed a technique to learn the mapping between low-level gist features and recorded eye movements during video gameplay. Judd *et al.* [53] combined low to high level features to model attention. They use horizontal lines detector trained from mid-level gist features as their mid-level features.

Goferman *et al.* [93] define a new interpretation for saliency, by introducing GT background regions (context) with a GT salient object based on image description. High-level semantic scene context is mostly under-explored for saliency detection. Liu and Han [94] proposed to use an existing scene classification network for extracting scene context features. Zhang *et al.* [9] encoded scene context by using a captioning network to capture the “major” objects in a scene, as shown in Figure 3.4. Incorporating high-level semantic scene context information would be interesting in exploring further approaches for top-down attention. It may benefit to capture detailed features of a scene to provide more information for saliency reasoning, rather than an overall representation of a scene as in [94]. Furthermore, it would also be useful to consider the semantic relationships between all the objects in a scene, instead of limiting to those “major” objects mentioned in the captions only [9].

3.4 Attention Mechanisms

Attention mechanisms have shown to be effective in improving natural language processing [95] and many vision tasks [96,97]. The attention mechanism discussed here is a computational process that learns to weight and force a network to focus more on useful features for the relevant task. This is typically done by learning new attention features (usually derived from

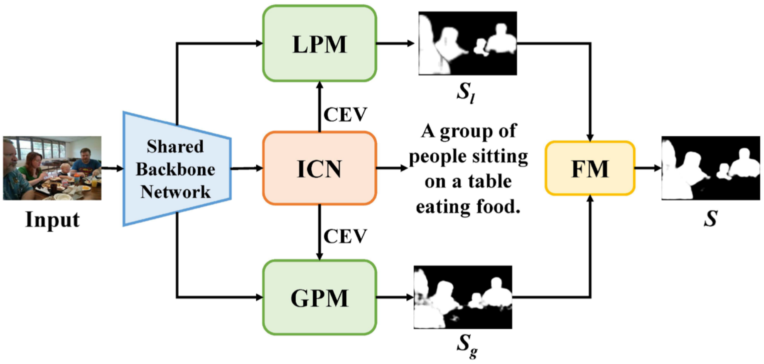


Figure 3.4: The CapSal model from [9], which leverages images captioning to extract high-level scene semantics for salient object detection. Image Captioning Network (ICN) encodes the generated caption (CEV) to capture the semantic information of major objects. Local Perception Module (LPM) exploits bounding boxes to capture local context for segmenting salient regions in a local view, and utilizes the encoded caption feature to boost classification of bounding boxes. Global Perception Module (GPM) incorporates the caption feature with global visual features to localize salient objects in a more global view for a holistic estimation. The Fusion Module (FM) generates the final saliency map (S) by fusing the saliency maps produced by LPM (S_l) and GPM (S_g).

initial contextual features) and multiplying with features relating to the task. The multiplication will raise values of certain elements in a vector or areas in an image. Raising the values of certain elements will then force a network to focus on those areas for further processing. It can be considered as top-down attention, since it guides the attention and focus of processing onto specific parts of information (e.g. , locations in an image) that relate to a task.

In salient object detection, attention mechanisms have been exploited to enhance multi-scale contextual features by capturing the interaction between pixels in local [98] and global contexts [25, 99–101]. Simple concatenation or element-wise operations on multi-level features may not improve saliency prediction [82] as noisy and non-relevant features may impact the saliency network [99]. To address this problem, [99] computes attention weights using convolutional layers on the local pixel neighbourhood. Zhang *et al.* [67] consider message passing to capture rich contextual information from multi-level feature maps and use a gating function to control the rate of message passing. Wang *et al.* [82] introduce a recurrent mechanism to gather multi-scale contextual information and iteratively refine convolutional features. Liu *et al.* [30] propose a Transformer-based network to unify RGB and RGB-D salient object detection. All these object saliency techniques apply attention mechanisms on region or patch-level features to find the most salient areas, while suppressing areas that do not contribute to saliency. It would be interesting to apply attention on the object-level and further integrate it with high-level scene semantics for saliency guidance.

3.5 Saliency Datasets

Initial datasets for salient object detection only contain one or two objects that generally have clear contrast from simple background. The MSRA-B [50] dataset contains images with mainly one salient object. Object segmentation data was manually drawn within the original user-labelled bounding boxes. The dataset includes variety of images from natural scenes, animals, indoor and outdoor. The SED [102] dataset presents two 100-image subsets, which only consist of images with one object or two objects.

Later works began to build new datasets with more challenging scenes and large number of images for training learning-based methods. SOD [103] is a dataset constructed from an existing segmentation dataset [104], which contains more than one salient object that is similar to background or near image boundary. The PASCAL-S [11] consists of 850 images that are more challenging than early datasets and ECCSD [10] compromises of 1000 images with semantically meaningful scenes. The DUT-OMRON [13] dataset contains 5168 images with up to five objects per image, while the HKU-IS [12] and MSRA10K [78] datasets contain over 4447 images and 10,000 images respectively. DUTS [14] is a large-scale dataset with 10,553 training images and 5019 test images. It's set of training images is primarily used by most saliency methods for training, while the test set and other smaller datasets are used for evaluation.

Though the later datasets are built to be more challenging, they still mostly consist of images with few objects in the foreground and background. The background areas are not always complex with similar visual textures to foreground salient objects. Introducing a new dataset with more foreground and background objects, with background also containing visual distractors similar to salient objects would be more challenging.

In terms of datasets for saliency ranking, [8] exploits the grey-scale values from the PASCAL-S [11] dataset to consider rank order. The authors further build a new dataset in their extended work [89][1]. They utilize the fixation maps from the SALICON [105] dataset and select ground-truth salient objects using the segmentation data from MS-COCO [27], with hand-designed criteria. Liu *et al.* [2] also introduce a new saliency ranking dataset by employing the fixation maps from SALICON and object segmentation from MS-COCO. Both datasets from [89] and [2] are based on the relative saliency ranking problem. Exploring attention shift as a definition for saliency rank order has not been explored, which can more closely reflect the natural behaviour of human visual attention.

[1] The dataset is not released and made publicly available.

Chapter 4

Problem Statement

Contents

4.1 Research Observations and Challenges **24**
4.2 Research Hypothesis **25**
4.3 Propsosed Ideas **26**

4.1 Research Observations and Challenges

Through the study of literature in saliency detection, we have observed potential weaknesses and limitations of recent saliency methods. These limitations and observations have produced interesting directions for saliency research.

Firstly, when we study early saliency techniques, they emphasise on applying bottom-up and top-down attention for saliency detection [40, 53, 60]. Current trend has shifted to using the power of deep-learning and CNNs, in which the connections between shallow and deep layers can be thought as imitating the bottom-up and top-down human attention process [6]. We observe that combining additional top-down approaches with deep-learning based methods is not fully explored.

We see that most of the existing salient object detection datasets [10–14, 103, 106] consist of images with few objects and simple backgrounds. Example images from these datasets are illustrated in Fig. 4.1. These images are relatively simple for salient object detection in the wild, where images are typically complex with lots of objects and complex backgrounds. With this observation, we realise that predicting saliency on complex images with rich context is crucial for advancing the performance and progress of salient object detection.

We find that many saliency work focus on improving the segmentation quality of salient objects on datasets containing images with only few objects. As the datasets are beginning to grow and become more challenging, we can look into new directions for saliency research. We find that saliency ranking is a relatively new problem with very few works investigating the task [2, 8, 89]. Saliency ranking goes beyond traditional salient object detection, as it also involves the problem of correctly rank ordering the salient objects. The rank order information can benefit subsequent tasks that require sequential processing and order of importance. For instance, the saliency rank order could be used to prioritize and plan robot navigation along areas of interest [107]. Another possible use case may involve as a recommendation system for design and visualization (e.g. , webpage), in which the saliency rank order can help designers build a web page that directs attention in a specific order [108]. We identify that salient object ranking based on attention shift has not been considered. Therefore, we determine it as an interesting avenue for research, especially since it mimics the human visual attention process more closely than the defined saliency ranking in [8, 89].

We notice that current salient object detection models are mainly trained on binary saliency labels that are class-agnostic. Training on such labels only can limit the ability of networks to learn semantic contextual features (higher-level understanding) that would otherwise help

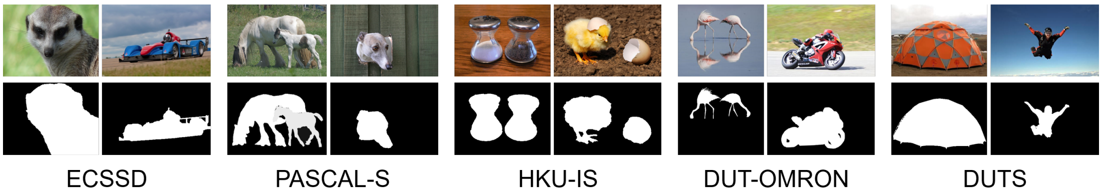


Figure 4.1: Example images from existing popular salient object datasets (e.g. , ECSSD [10], PASCAL-S [11], HKU-IS [12], DUT-OMRON [13] and DUTS [14]). Current saliency datasets are not very challenging and contain very few objects with simple background. Top row shows sample images from different datasets and the bottom row show corresponding ground-truth saliency.

model various relationships of objects within complex image scenes [9, 94]. We also observe that there is little work that incorporate high-level semantic context information for improving saliency reasoning. The works in [94] and [9] exploit high-level scene context, however, they limit their scene context to abstract representation and captioning of major objects only.

Another observation is that many of the existing state-of-the-arts are patch-based methods [22, 26, 65, 69], which may only capture parts of salient features and not the whole object. This is not optimal as information of whole objects are not propagated and using only partial features may mislead accurate saliency prediction. Furthermore, patch-based methods are unable to obtain salient object features as individual object instances, which could also affect the saliency decision making between multiple objects. We believe that capturing salient features on the object-level would enhance the representation of multiple salient objects and boost the saliency reasoning between them.

4.2 Research Hypothesis

Based on the observations of current limitations in saliency and potential research directions, we make several hypotheses:

1. Our hypothesis is that we can build a dataset for investigating salient object ranking based on attention shift, by utilizing the temporal information of fixation data. We believe that it would be useful for developing saliency ranking models for emulating shifts in human visual attention.

2. We hypothesize that deep learning techniques will be able to learn and predict salient object ranking based on attention shift.

3. We also hypothesize that combining bottom-up and top-down approaches would be an effective technique for ranking salient objects.

4. Our hypothesis is that we can efficiently build a large-scale salient object detection dataset consisting of challenging images. We also believe that it would help test the performance of saliency methods.

5. We hypothesize that we can learn and use discriminative semantic context to improve saliency modelling in challenging complex scenes with rich context.

6. We hypothesize that capturing individual salient features on the object-level would boost saliency reasoning.

4.3 Propsosed Ideas

We carry out studies and experiments to test the above hypotheses.

We begin by experimenting with various methods for generating ground-truth salient object ranking based on attention shift (**H1**). We perform a user study to select and justify the best method for generating our ground-truth rank data. We then propose a novel deep-learning model for learning to rank salient objects on our new dataset (**H2**). We combine various bottom-up and top-down approaches to design an effective salient object ranking model (**H3**). This work is detailed in Chapters 5 and 6.

Next, we explore traditional salient object detection on challenging images scenes. We first construct a new large-scale salient object detection dataset with complex image scenarios (**H4**). This is done by an automatic and manual ground-truth generation phase, based on findings from previous work and our analysis during the data generation process. Subsequently, we propose a deep-learning saliency network that leverages panoptic segmentation for extracting detailed semantic context of a scene (**H5**). This work is showcased in Chapters 7 and 8.

We exploit object detection for generating object proposals, thus, producing object-level feature representation for individual instances. More specifically, we adopt an existing object detection framework that provides detection of individual objects and their segmentation. We build our saliency models on top of the modified object detection framework with low and high -level features. Not only does this provide object-level features to improve saliency prediction (**H6**), but it also offers an alternative approach for bottom-up and top-down saliency (**H3**). This is showcased in Chapters 5, 6, 7 and 8.

Chapter 5

Saliency Ranking

Contents

5.1 Introduction **28**

5.2 Proposed Dataset **31**

5.2.1 Data Generation 31

5.2.2 User Study 34

5.2.3 Dataset Analysis 36

5.3 Proposed Method **37**

5.3.1 Network Architecture Overview 37

5.3.2 Backbone Network 38

5.3.3 Selective Attention Module (SAM) 39

5.3.4 Spatial Mask Module (SMM) 40

5.3.5 Saliency Rank Network 41

5.4 Experiments **42**

5.4.1 Comparison with State-of-the-Arts 43

5.4.2 Ablation Study 45

5.4.3 Further Comparison with S4Net 47

5.4.4 Saliency Ranking on Different Contexts 47

5.5 Limitations and Future Work **48**

5.6 Summary **49**

5.1 Introduction

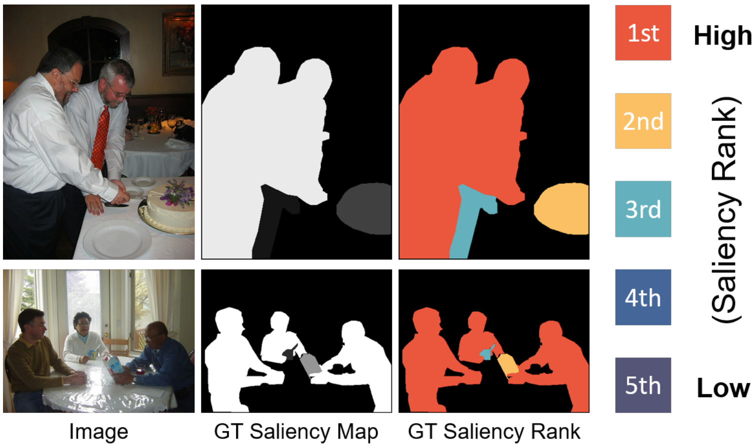


Figure 5.1: Sample images from PASCAL-S dataset [11], which is used for saliency ranking in [8]. Note that multiple object can be given the same rank. The colours (orange→purple) indicate the saliency rank 1→5.

As previously mentioned in Chapters 1 and 4, many saliency work have been performed on salient object detection as a binary prediction problem [6, 22, 23, 26, 80]. Due to the limited capacity of the human visual system, we humans have difficulties in processing multiple simultaneous visual system at once [45]. Accordingly, we have adapted to develop a sequential visual process of selecting and shifting our attention from one object to another [41, 42]. Very few works explicitly model human attention shift from one object to another.

Both [109] and [110] employ the *gaze-following* concept to find objects or regions that are likely gazed by humans. They incorporate a gaze-pathway that takes human head regions and locations to generate a mask. The mask indicates the likely locations that humans would gaze towards. Combining with a saliency map, they produce the final gaze saliency. The two works are limited to only social scenes and does not explore attention shift among multiple generic objects. It would be more challenging to investigate attention shift among general object classes, as objects that influence attention shift may not present especially when there is limited interaction among the objects in a scene.

Islam *et al.* [8] introduce the problem of relative ranking of salient objects on an existing PASCAL-S dataset [11]. The relative rank is inferred from the agreement of binary object saliency among multiple observers. The study is motivated by the fact that observers are likely to have different views of what objects are considered salient. In their implementation,

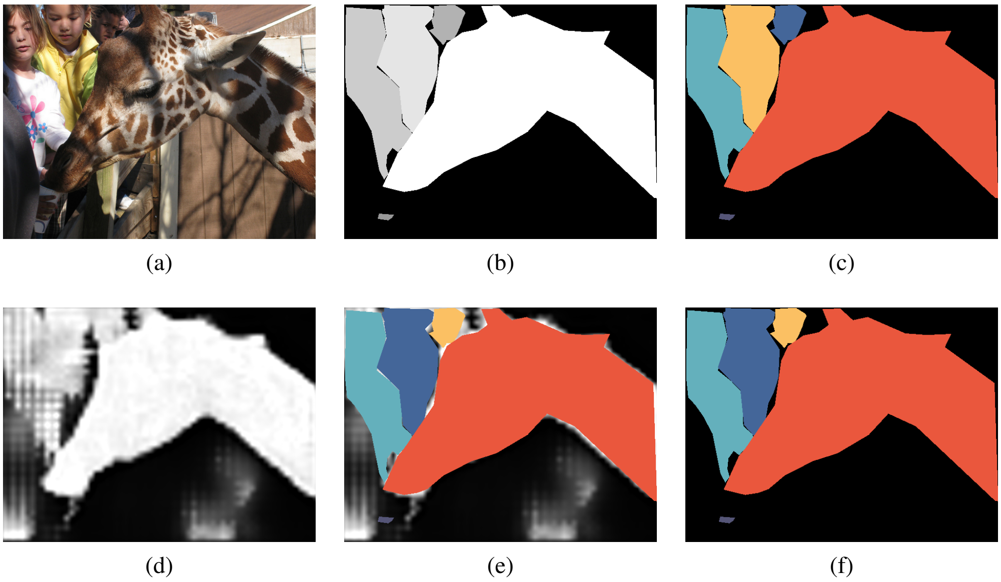


Figure 5.2: (a) Example input image, (b) corresponding ground-truth (GT) saliency rank, (c) corresponding GT saliency rank (colourised), (d) saliency rank prediction by RSDNet [8], (e) corresponding saliency rank by RSDNet (GT objects overlaid and colourised) and (f) corresponding saliency rank by RSDNet with only GT objects (overlaid and colourised). Note that the accuracy of predicting correct ranks can be impacted by the quality of the objects captured.

they implicitly assume that multiple objects picked by the same observer share equal saliency rank (Figure 5.1). Simultaneous attention to multiple objects, however, is not supported by behavioural observations because dividing attention between multiple objects often leads to poorer performance [47] and may not truly reflect how humans shift their attentions. Multiple objects with the same rank would also make it difficult to model the order of attention shift.

Inspired by the aforementioned saliency and psychological studies [45], we aim in this chapter to investigate saliency rank that models human attention shift. We first propose a new saliency ranking dataset collected based on attention shift (Section 5.2). Our idea follows psychology studies that humans attend one object at a time in a complex scene [41]. We consider that the first object attended by an individual should have the highest saliency. Subsequent attended objects should be associated with descending saliency values (i.e. , attention shift towards objects of lower saliency values). Since different observers may have different saliency ranks on objects, we take the average of the saliency ranks from multiple observers to obtain the ground-truth saliency rank (Section 5.2.2). We show, with a user study, that such human attention shift on object instances correlates with object saliency rank.

Traditional saliency models often introduce many false positive saliency to non-salient objects and background (Figure 5.2(d-f)). When the shape of the objects is not captured well (Figure 5.2(d)), it further impacts the saliency rank prediction of the objects (Figure 5.2(f)). Motivated by the above observations, we propose a saliency rank prediction method to infer human attention, leveraging both bottom-up and top-down attention (Section 5.3), resulting in state-of-the-art ranking performances (Section 5.4). Our model carries out object proposal, object segmentation and object rank prediction in one go, allowing our network to reason saliency ranking on the object-level and enables the capture of individual salient instances, while most prior works (e.g. , [8]) perform at region-level and make no object proposals.

Contributions: In this work, we focus on salient object ranking based on attention shift. Our main research question is: *Can we combine bottom-up and top-down approaches for learning to rank objects based on attention shift?* The main contributions of this work include:

- We propose a new research problem to predict salient object ranks according to human attention shift. It is inspired by psychological and behavioural studies. It goes beyond just human-object interactions [109], and can also model object-object attention shift.

- We propose a new large-scale dataset for the problem of salient object ranking, justified by our user study.

- We propose a deep learning approach to jointly predict saliency ranks of multiple salient object instances and their corresponding object masks, with bottom-up and top-down attention mechanisms.

- Extensive experimental evaluations and analyses show that the proposed model achieves state-of-the-art performances on salient object ranking compared with relevant methods.

The findings, results, code and data from this chapter were published in [1][1].

The rest of this chapter is structured as follows: Section 5.2 explains the approaches we consider for generating saliency rank dataset, user study to determine the best approach and the analysis of the final dataset. We then detail our proposed method in Section 5.3 and evaluate our method in Section 5.4. Finally we discuss limitations in Section 5.5 and conclude in Section 5.6.

[1] Avishek Siris, Jianbo Jiao, Gary K.L. Tam, Xianghua Xie and Rynson W.H. Lau. “Inferring Attention Shift Ranks of Objects for Image Saliency”. Proceedings of the *IEEE/CVF Conference on Computer Vision and pattern Recognition (CVPR)*, 2020.

5.2 Proposed Dataset

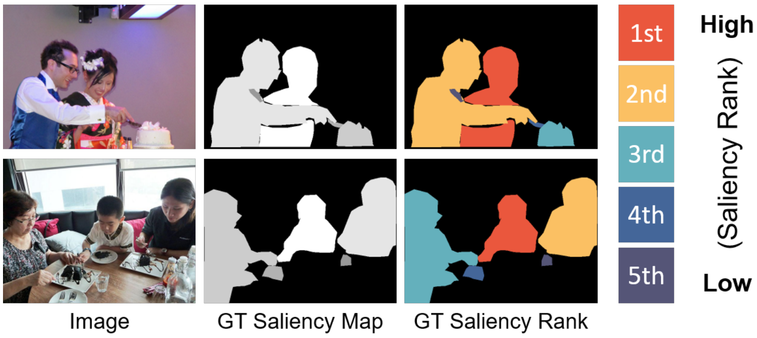


Figure 5.3: Sample images from our proposed dataset. Our dataset contains images with a maximum of 5 unique saliency ranks.

5.2.1 Data Generation

To our knowledge, there are no large-scale datasets available for salient object ranking based on *attention shift*. Hence, we build a new large-scale salient object ranking dataset (examples shown in Figure 5.3), by combining the widely used MS-COCO dataset [27] with the SALICON dataset [105]. MS-COCO contains complex images with ground-truth object segmentation, while SALICON is built on top of MS-COCO to provide mouse-trajectory-based fixations. The SALICON dataset provides two sources of fixation data: 1) fixation point sequences and 2) fixation maps for each image. A fixation point is the area where an observer is focusing there visual attention. Usually this is captured by an eye-tracking device that records eye movement data. SALICON instead capture fixation points using a mouse-based system. The position of the mouse relates to the fixation point and the movement of the mouse corresponds to the eye movements (more details can be found in [105]). The fixation map is produced by applying a Gaussian filter onto the fixation point data. This aggregates the fixations and yield a heat-map like fixation (saliency) map.

We exploit the two sources of fixation data and consider three approaches to build our ground-truth saliency rank annotations. The first approach awards higher saliency values to objects fixated early in a fixation sequence. The second approach focuses only on the order of distinct objects that were fixated without repetition. The third approach uses the pixel intensity values from a fixation map. Both the first and third approaches are further extended into four methods each. In total, we consider nine methods to generate possible ground-truth annotations, which we will discus below. We do not know which methods would reflect the way that humans rank multiple objects in term of saliency. We carry out a user study in Section 5.2.2, and provide some analysis on our dataset in Section 5.2.3.

We consider up to top-10 objects in the user study, but use top-5 for saliency ranking prediction. We believe that top-5 ranks is a good setting. Top-5 ranks contain clear and easy to define ranks of the top-5 objects. It is challenging but do not contain too many ranks, where the saliency differences among the lower ranks (ranks > 5) sometimes becomes minuscule and unclear. Having too many ranks would be too difficult for saliency rank prediction, which is an unexplored problem.

Approach 1: For each image, we follow the fixation points in a fixation sequence and assign descending saliency scores to the fixated image pixels. We repeat this scoring of pixels over all observers' fixation data. The saliency rank of an object can be computed by aggregating these saliency scores that the object contains (i.e. , the higher the aggregated score, the more salient the object and the higher the rank). The number of fixation points varies among observers, leading to a large difference in scores.

We first assign scores to pixel values using fixation points from the SALICON [105] dataset. We then obtain the score for each object based on the values of pixels belonging to the object. Specifically, for every image $I \in \mathbb{R}^{W \times H}$ of dimension $W \times H$, there are N observers. Let F^j be the fixation sequence obtained from one of the N observers $j \in [1,N]$ and a fixation f_i^j with index order $i \in [1,t]$ that represents the i^{th} fixation in the sequence F^j of length t. We assign a score to image pixel p if the fixation f_i^j falls on p using:

$$v_p = \sum_{j}^{N} \sum_{i}^{t} g(f_i^j), \quad \text{if } f_i^j = p, \tag{5.1}$$

$$g(f_i^t) = 1 - \frac{i}{t}, \tag{5.2}$$

where v_p denotes the score of pixel $p \in I$ aggregating from all N observers' fixation data. Function g takes the temporal order i^{th} of a fixation point in the sequence into account, and assigns a lower value to a fixation point if it is latter in the sequence.

To build our datatset on the idea of attention shift for saliency ranking, it is important for us to focus on the order of fixation points. By doing so, we will be able to closely define ground-truth saliency rank based on the sequential shift of attention, as observed by humans [41]. We thus do not take into account the duration of the fixation points in our formulation for two main reasons. First, there are large variances in the duration of fixations among different observers. Second, it is difficult (if not impossible) to obtain the exact duration of each fixation point whilst the fixations are obtained from a re-sampling process [105]. In contrast, using the order of fixation points would ensure that there is a consistent gap between the scores of each pair of consecutive fixation points, leading to a higher stability in the final object scoring.

Next, we try to accommodate the varying sizes of objects in an image. Larger objects may collect more fixations from observers and be considered more salient with higher ranks. However, small objects that are rare may also be more salient even if there are fewer fixations. As we are unsure which methods best model how human ranks, we develop four methods to aggregate scores for subsequent object saliency ranks, namely: (1^{st}) *FixSeq-avg* (average score), (2^{nd}) *FixSeq-max* (maximum score), (3^{rd}) FixSeq-avgPmax (average + maximum score) and (4^{th}) *FixSeq-avgMmax* (average × maximum score). Let o be one of the objects (from MS-COCO ground-truth annotation data) in an image I, $|o|$ be the number of pixels in o, and v_p^o be the score of a pixel $p \in o$ inside an object. We define:

$$\textit{FixSeq-avg}(o,I) = \frac{1}{|o|} \sum_{p \in o} v_p^o, \tag{5.3}$$

$$\textit{FixSeq-max}(o,I) = \max_{p \in o}(v_p^o), \tag{5.4}$$

$$\textit{FixSeq-avgPmax}(o,I) = \textit{FixSeq-avg}(o,I) + \textit{FixSeq-max}(o,I), \tag{5.5}$$

$$\textit{FixSeq-avgMmax}(o,I) = \textit{FixSeq-avg}(o,I) \times \textit{FixSeq-max}(o,I). \tag{5.6}$$

For a given image, *FixSeq-avg* (Eq. 5.3) calculates the final score of an object by taking the average values of pixels belonging to the object. It takes into account the size differences between objects. In *FixSeq-max* (Eq. 5.4), the final score of an object is the maximum value v_p^o of all its pixels. It ranks objects higher if they are observed earlier in the fixation sequence. It does not consider the object size. For the methods *FixSeq-avgPmax* (Eq. 5.5) and *FixSeq-avgMmax* (Eq. 5.6), we consider weighting the final scores by performing addition or multiplication with the results from Eq. 5.3 and Eq. 5.4, respectively. The use of addition in *FixSeq-avgPmax* is a shorthand of averaging the effect of both *FixSeq-avg* and *FixSeq-max* values. *FixSeq-avgMmax* considers to weight *FixSeq-avg* by multiplying *FixSeq-max*. We then sort all objects in descending order of the saliency score, and each object is given a distinct rank.

Approach 2: This approach also considers temporal order. However, we only focus on the first T distinct objects and ignore repeated fixations on already visited objects. In addition, we directly assign a score to the whole object if a fixation point resides in its segmentation. We term this method as (5^{th}) *DistFixSeq*. Specifically, we define a new sequence $\hat{f}_i^n$ by removing fixations that fall on objects already visited by earlier fixations in f_i^n. We then define

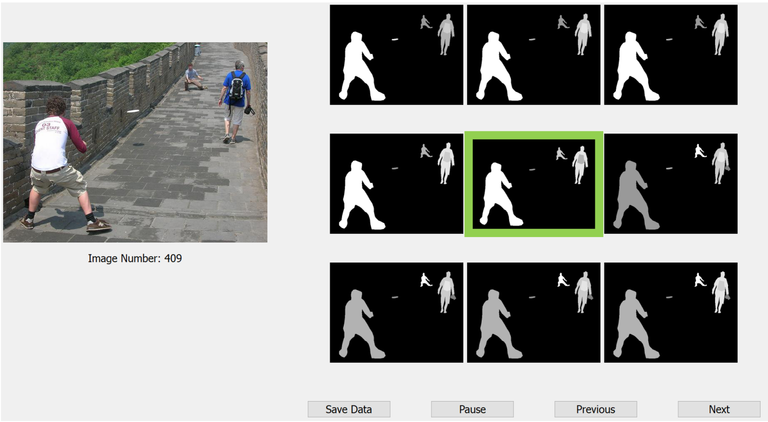


Figure 5.4: Screenshot of the annotation tool used by the participants during the user study. Participants are not told how the maps are generated. They are asked to pick a map that best respects the "order of attractiveness". The green box indicates the map picked by one of the participants.

DistFixSeq, for each object o in an image I, as:

$$DistFixSeq(o,I) = \frac{1}{N}\sum_{j}^{N}\sum_{i}^{T} h(\hat{f}_i^j), \quad \text{if } \hat{f}_i^n \in o, \tag{5.7}$$

$$h(\hat{f}_i^n) = T - i, \tag{5.8}$$

where $T = 10$. Function h assigns higher scores to objects if they are observed earlier. Eq. 5.7 takes into account only the first T objects and averages the final scores across all N observers. We obtain the object ranks in the order of descending scores.

Approach 3: We use the fixation maps in this approach as the source for the saliency score. We directly take intensity values from the fixation map as pixel scores v_p. Similar to ***Approach 1***, we define four methods to generate the final scores for each object. Accordingly, we have (6^{th}) *FixMap-avg* (average score), (7^{th}) *FixMap-max* (maximum score), (8^{th}) *FixMap-avgPmax* (average + maximum score) and (9^{th}) *FixMap-avgMmax* (average × maximum score). These four methods compute the final object scores in the same way as their counterparts in ***Approach 1*** (i.e. , Eq. 5.3-5.6). Again, we consider the first distinct T objects, and assign the saliency rank in the order of descending scores.

5.2.2 User Study

We perform a user study with 11 participants to find out which of these nine methods produce more consistent ground-truth attention shift order based on human judgment. For each image, the participants were presented with the image and the nine corresponding saliency rank maps

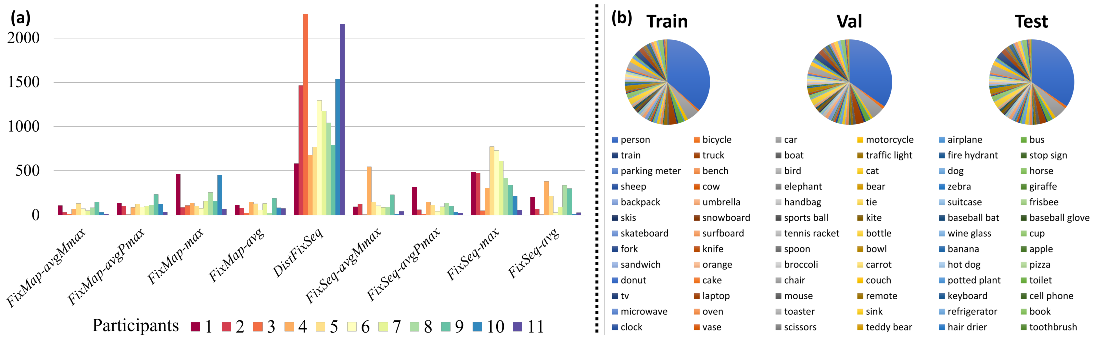


Figure 5.5: (a) Pick rates of maps by 11 participants in our user study across 2500 images. These maps are generated by nine methods that we experimented with in Section 5.2.1. (b) Distribution of ground-truth salient instances of all object categories in each data split of our dataset.

arranged in a grid. Figure 5.4 shows an example screenshot of the annotation tool used in the user study. After a briefing session on how to use the annotation tool, each participant is told to observe the image first, and then pick a map that show objects with “order of decreasing attractiveness”. Participants are not told how the maps are generated. Each participant was asked to annotate a set of 2500 images. These images are randomly sampled from our dataset. Participants annotate them in 5 sessions (500 images each). Each annotation session lasts under an hour on average. After the annotation task, participants were rewarded with a £25 Amazon gift voucher.

Figure 5.5(a) shows that, on average, the map generated by (5^{th}) *DistFixSeq* has the highest number of picks. The map aligns most to the order of attractiveness of objects. It suggests that the temporal order of fixated objects (attention shift) is vital for determining the strength of attractiveness among multiple objects. Attractiveness of objects is considered as attracting attention towards the objects and reflects their saliency strength [111].

We can further see that there are more picks of the methods from ***Approach 1*** (maps generated from temporal fixation) than those from ***Approach 3*** (maps generated from fixation map only, without temporal data). This suggests that ignoring the temporal fixation order, or using the order by fixation intensity alone, does not always capture the expected order of saliency (attractiveness of objects). These results correlate to the idea of attention shift by descending saliency values in [42], and prompt our definition of saliency rank order via attention shift. It supports us to use (5^{th}) *DistFixSeq* to generate the ground-truth saliency ranking for the development of our rank prediction technique.

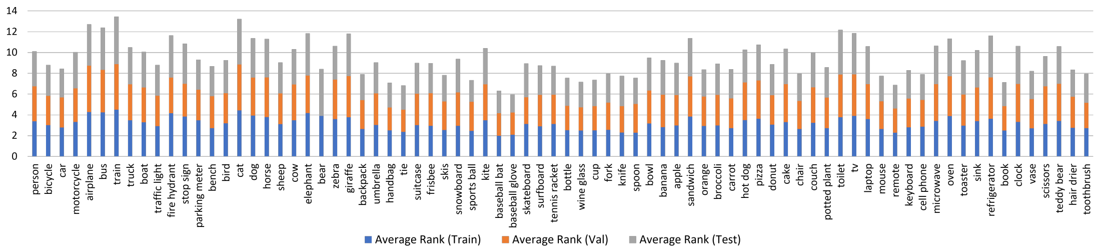


Figure 5.6: Average rank of each object category in the proposed dataset.

5.2.3 Dataset Analysis

Our dataset is adapted from MS-COCO [27] and SALICON [105], and thus share similar characteristics. All existing popular datasets (e.g. , ECSSD [10], DUTS-OMRON [13], PASCAL-S [11], HKU-IS [12], DUTS [14]) target binary salient object detection while ours focuses on **salient object ranking**. Our dataset contains more complex images and is the largest in size. Note that all other datasets do not include individual object labels, making them ill-suited for our task.

We report that the average number of objects per image in our dataset is around 11 (maximum of 68). The "person" object category occurs most frequently in the dataset. This is expected as most photo images target people as the subject. Additionally, many images contain crowd of people with small individual annotations, causing the total count to be 4-16 times greater than other categories. Correspondingly, "person" objects receive the most instances of ground-truth saliency, which aligns with previous observations that humans usually attract attention [53]. Figure 5.5(b) shows the distribution of ground-truth salient instances of each object category in our dataset. Figure 5.6 shows the average rank of each object category based on instances, given the ground-truth saliency. From Figure 5.6, we can see that large objects (e.g. , "train", "airplane") have fewer instances per image, and some animal categories (e.g. , "cat", "dog", "elephant") have a larger rank average score than the "person" object. We also find that object categories relating to appliances (e.g. , "refrigerator", "microwave") have quite high scores, which mainly come from indoor scenes with no other object(s) of interest.

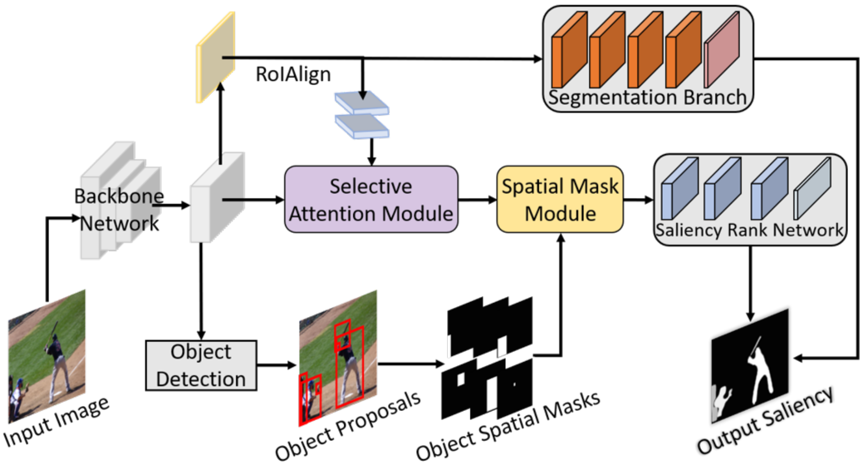


Figure 5.7: Architecture Overview. The model consists of a backbone network, Selective Attention Module (SAM), Spatial Mask Module (SMM) and a classification network for salient object ranking. We utilise Mask-RCNN [15] as our bottom-up backbone to provide object proposals with the FPN [16], and object segmentation from the segmentation branch. The bottom-up SMM extracts low-level features of the proposed objects while the top-down SAM considers high-level contextual attention features.

5.3 Proposed Method

5.3.1 Network Architecture Overview

We propose a CNN model to predict saliency rank with a bottom-up bias stimuli [112, 113], which we find useful to pick up the most salient objects in the scene. The saliency rank, especially on those less salient objects, may relate to the scene structure and observer interpretation [114]. As a result, the saliency rank modelling requires higher-level cues and prior knowledge [115].

The proposed network architecture consists of four modules, namely, a backbone network based on Mask-RCNN [15], a Selective Attention Module (SAM), Spatial Mask Module (SMM) and a saliency rank network, as illustrated in Figure 5.7. They are arranged to provide alternate bottom-up and top-down attention mechanisms.

Mask-RCNN generates object proposals as a bottom-up approach similar to [116]. This provides us individual object features and allows us to learn semantics information on the object-level in subsequent modules. Next, the SAM compares the features of each object to the global semantic image features in order to determine relevant target salient objects. This module provides a top-down attention mechanism and is motivated by psychophysical findings that humans frequently gaze towards interesting objects. It encapsulates important scene semantics [117] and interpretation due to eye gazes [114]. We then combine the features output by SAM with spatial masks in the SMM. We use spatial masks as a low-level cue, which embeds the relative size and location of each object in the image. Finally, we infer saliency rank of object instances with a small classification network. We adopt the segmentation branch of Mask-RCNN to produce segmentation for the object instances.

5.3.2 Backbone Network

Objectness and object proposals for binary salient object detection have been explored in [118–120]. Feng *et al.* [118] extend the global rarity principle (rare and less frequently occurring objects are likely to be salient) to derive object saliency. It uses a sliding-window mechanism to determine if the features inside the windows contain foreground or background features. [118] and [120] further extend it to many sliding windows of various scales. Fan *et al.* [17] present a model architecture much like the Mask-RCNN [15]. They produce object proposals by adopting the Feature Pyramid Network (FPN) [16] and propose a salient instance segmentation branch that extends the segmentation branch in Mask-RCNN. The purpose of their network is to perform salient-instance segmentation, while we investigate salient object ranking based on attention shift order.

Inspired by these work, we adopt Mask-RCNN as the backbone of our model and to provide efficient object proposals and segmentation. The FPN serves as a bottom-up attentive mechanism [116].

To model saliency in the object-level, we apply RoIAlign [15] and two fully connected layers (FCs) to extract object-level features, $o_i \in \mathbb{R}^{1024}$, for each object proposal, leading to a set of object features $O = \{o_1, o_2, \ldots, o_M\}$, where $M = 30$ is the maximum number of object proposals. We further take the pyramid features "$P5$" from the FPN as the high-level features input to the SAM module for top-down attention. The segmentation branch generates pixel-wise segmentation of objects for a clearer final saliency map. Different from [118–120], we do not output bounding boxes of salient objects. Instead, we predict a saliency map that indicates the pixel-wise segmentation and the saliency ranks of object instances. In contrast to [17], we exploit components of Mask-RCNN to build our bottom-up and top-down model for salient object ranking.

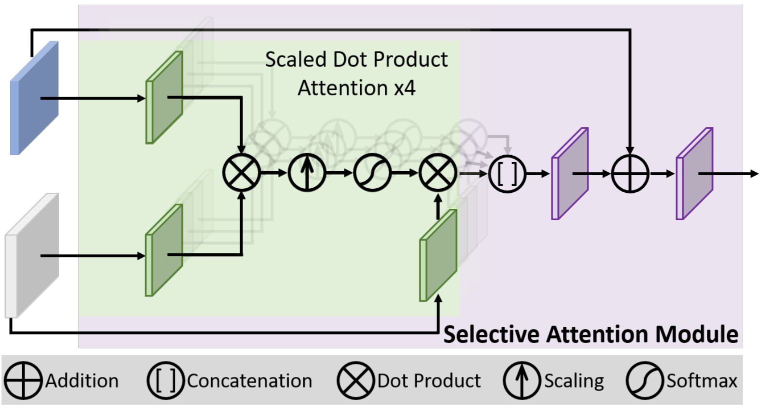


Figure 5.8: Details of the Selective Attention Module (SAM). Blue Block: object feature. Grey Block: pooled image feature. Green and Purple Blocks: fully connected layer applied onto corresponding features. Each block is a 512-D feature vector.

5.3.3 Selective Attention Module (SAM)

A straightforward choice to model how humans attend one object to another would be a recurrent strategy. Such a strategy is computation and memory intensive, especially when there are a lot of objects in an image (like those in our proposed dataset). To model all relationships of objects and their associated attention shift probabilities in a potential sequence, it would easily lead to an exponential growth problem as the number of proposals increases. Instead of using recurrent strategy to model attention shift, we get inspirations from recent task-based techniques [95,96,121–124], which were greatly benefited from some forms of attention mechanisms. These mechanisms are often designed to dynamically weight relevant features or entities tailored to certain tasks while suppressing the distractors. Here, we consider that an attention mechanism would be useful to infer the way observers shift their attentions because it encapsulates important scene semantics [117] and interpretation due to eye gazes [114]. In addition, though human actors in an image would affect observers to shift their gazes [110], we consider that individual generic objects may not necessarily have such strong influence on attention shift. For generic images (e.g. , non-human scenes and images with little interactions among objects), we consider that the scene structure and relationship between objects may have a stronger influence on attention shift [61]. We thus develop a Selective Attention Module (SAM) to compute top-down attention by comparing object features individually to the image scene features.

We build the attention module using Scaled Dot-Product Attention [95] (Figure 5.8) with image and object features. We use the pyramid features, "$P5$", from the backbone network as the image features. A (1×1) convolution and global average pooling are applied onto the pyramid features to obtain our high-level image representation.

Before computing the dot-product, we first project the object and image features into a 512-D space [95]. Here, we embed the features of each object into an individual feature vector using a shared FC layer. Two separate feature vectors are generated with separate FC layers, both taking the pooled image features as input. The sets of new features from the pooled image features are further repeated M times. The attention mechanism then use these embeddings to perform dot product similarity of individual object features with the image features. We add scaling factor [95], and apply softmax activation to obtain the attention score. Our attention module computes attention scores with multiple heads (4 heads) in parallel. The idea is that each attention head would learn different high-level information to guide scoring/weighting for salient targets. The outputs from multiple attention heads are concatenated and then sent through a FC layer. Finally, we add a residual connection and a FC layer for the module output.

5.3.4 Spatial Mask Module (SMM)

Understanding the relationship between object properties and scene context can help select relevant targets in a complex scenario [125]. For example, very small objects in a scene may not attract human attention. Objects close to the centre of the image may be more salient due to the "center bias" concept [13, 53]. These motivate us to include low-level objects properties (e.g. , size and locations) to learn contextual features that model relationship between objects and scene.

Using the bounding boxes of object proposals, we generate a spatial mask for each object. Spatial masks embed the size and location of the proposed objects in relation to the visual scene. We capture such information with a binary mask (i.e. , assigning a value of 1 to pixels within a bounding box, and 0 otherwise). We pass the spatial masks through three convolutional layers to compress each of them into a 64-D feature vector (detailed in Figure 5.9). Each set of spatial features is then combined with the corresponding object features via a concatenation layer. It is similar to the procedure of embedding positional information in the Transformer [95] before the attention. This module can be considered as a process of combining bottom-up and semantic attributes of objects [117].

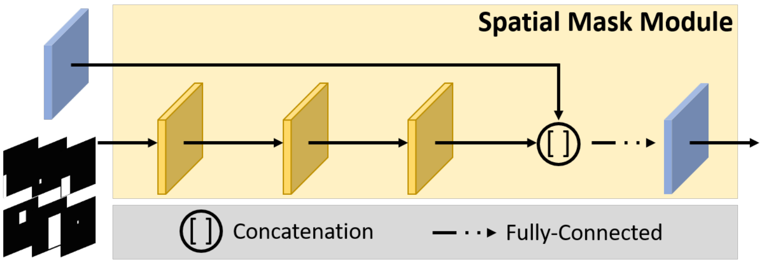


Figure 5.9: Details of the Spatial Mask Module (SMM). Blue Block: 512-D object feature vector. Yellow Blocks: convolutional layers with output feature dimensions (96, 128, 64), kernel size (5×5, 5×5, 8×8) and strides (2, 2, 1). The three convolutional layers outputs a 64-D feature vector from an input 2D map (32×32) that represents the spatial information for each object.

5.3.5 Saliency Rank Network

Our initial attempt to model salient object detection and attention shift order ranking is to cast it into a classification problem. In our setting, we consider $C = 5$ ranks and leave exploring higher ranks as future work. With one additional background class for non-salient objects, our classification has $6 = 5+1$ classes. Saliency and rank are then predicted with a small classification network consisting of three convolutional layers and one classification layer. During inference, we combine the saliency rank classification with object segmentation (from the segmentation branch) to generate the final salient object rank map. However, a classification formulation cannot ensure that the detected salient objects would be assigned distinct saliency ranks. To address this problem, we instead use the softmax rank classification probabilities in a scoring mechanism. For each object, we first take the probability of its predicted saliency rank as the initial score. We then add and multiply the initial score with a value relative to the predicted rank. The final rank score for each object is given by the following:

$$r = (r_p + r_c) \times r_c, \tag{5.9}$$

where r is the final rank score, r_p is the softmax rank classification probability and r_c is a score relative to the predicted rank class for an object. Objects that are supposedly of higher ranks will accumulate higher scores. This is inspired by [8], which determines object saliency rank by the descending average pixel saliency value of each object. By doing so, we can ensure distinct saliency rank to be predicted for each object.

We consider the top-5 saliency rank order of objects from their descending score values. We take the top-5 available objects with the highest scores and assign them corresponding top-5 discrete ranks based on their score order. Finally, we generate a saliency map by combining the predicted segmentation maps of the top-5 available salient objects. We indicate the ranks of each salient object by assigning unique grey-scale values (based on their rank) to their segmentation.

5.4 Experiments

This section explains the settings for conducting the experiments and evaluating the performance between our proposed method and state-of-the-arts.

We fine-tune our backbone components of Mask-RCNN on salient objects before training our final model on salient object ranking. A pre-trained ResNet-101 [126] is used to initialise the convolutional layers of the Mask-RCNN. All images during training and testing are resized to 1024 × 1024 before feeding into the network. During inference, we resize the output saliency map back to the original size of 640 × 480. Our model is implemented by the Tensorflow framework and trained on an Nvidia GTX 1080 Ti GPU. We set the mini-batch size to 8. We train variations of the network for 40 epochs each, taking a maximum of 6 hours for one model training. We use the SGD optimizer with gradient norm clipping set to 5. Learning rate is set to 10^{-3}, with momentum and weight decay configured as 0.9 and 10^{-4}, respectively.

Our dataset employs the same set of images and fixation sequence from SALICON [105], and contains object segmentation masks from MS-COCO [27]. The SALICON dataset consists of 10K training, 5K validation and testing images. There are no annotations for the test set. We use the training and validation sets to build our dataset. We consider saliency ranking based on the fixation sequence of the first 5 distinct objects visited without repetition (*DistFixSeq*, Section 5.2). The choice of the method is supported by our user study. We discard images with no object annotations, and those images containing smaller objects that are completely enclosed by larger ones. Finally, we use images containing at least two salient objects (i.e. , at least two ranks) to ensure that we have attention shift for our salient object ranking task. The dataset is randomly split into 7646 training, 1436 validation and 2418 test images, respectively.

We use the Salient Object Ranking (SOR) metric [8] for evaluation. It is formulated as the Spearman's Rank-Order correlation between the rank order of the predicted salient objects and the ground-truth. The correlation ρ is computed as:

$$\rho = 1 - \frac{6\sum d_i^2}{n(n^2-1)}, \tag{5.10}$$

where d_i is the rank difference between the corresponding object pairs in the predicted and ground-truth rank sets. n is the number of objects paired in both sets (e.g. , up to 5 ranked objects in our case). The metric measures the strength and direction of the monotonic relationship between two rank order lists with $[-1,1]$ indicating negative to positive correlation. However it does not cater for the case when there are no common objects between the two rank variables. For example, when one technique predicts a completely different set of objects from the ground-truth, SOR is not defined. Therefore, we further report how many images were used to calculate the average SOR for the whole test set, where the more images used the more reliable the SOR is. The reported SOR measurement is all normalised to [0,1].

We also do a comparison with the mean absolute error (MAE), which measures the average per-pixel difference between the prediction and ground-truth. It is defined as:

$$MAE = \frac{1}{W \times H} \sum_{x=1}^{W} \sum_{y=1}^{H} |S(x,y) - G(x,y)|, \tag{5.11}$$

where $W \times H$ is the image dimension, S and G are the predicted and ground-truth saliency maps, respectively. We calculate MAE between the original predicted saliency map and the ground-truth map, before any post-processing of saliency prediction to obtain the saliency rank. It is an alternative measure for the quality of both predicted saliency maps and ranks. It also works even when a technique predicts a completely different set of objects from the ground-truth.

5.4.1 Comparison with State-of-the-Arts

Table 5.1: Comparison with state-of-the-art methods on our dataset. Note that RSDNet scores are based on direct prediction with pre-trained weights from their dataset. ↑(↓) means the higher(lower) the better. Top two scores are shown in red and blue, respectively.

Method	MAE ↓	SOR ↑	#Images used ↑
RSDNet [8]	0.139	0.728	2418
S4Net [17]	0.150	0.891	1507
BASNet [18]	0.115	0.707	2402
CPD-R [19]	0.100	0.766	2417
SCRN [20]	0.116	0.756	2418
Ours	0.101	0.792	2365

Quantitative Evaluation: We compare against five state-of-the-art methods, namely the RSDNet [8], S4Net [17], BASNet [18], CPD-R [19] and SCRN [20], in which RSDNet first introduces saliency rank. Note that all these methods do not predict object segmentation and instead only provide a single binary saliency map.

The S4Net is chosen since it has a similar structure to our backbone and outputs object instance segmentation. We modify the S4Net code in order to predict up to 6 classes for each object instead of the binary prediction as in their original paper [17], for a fair comparison. We then apply our method of inference to obtain distinct saliency ranks. For all the rest compared models and RSDNet, the predicted saliency ranks of ground-truth objects is obtained by averaging the saliency value of pixels that belong to each object. Discrete object ranks is then determined and assigned by descending order of such averages.

The experimental results are shown in Table 5.1, which shows that our method outperforms other methods on the proposed dataset, achieving the best overall performance with better scores among all measurements (MAE, SOR and Images used). Note that RSDNet uses all images during the SOR calculation, due to its single binary saliency maps often containing many false saliency. Noise or very weak saliency is often propagated throughout the image and reach parts of the objects. This allows RSDNet to obtain saliency rank by averaging object pixel values to cover most objects.

S4Net shows the highest SOR score; however, it is only able to calculate the score in under two thirds of the test images. The rest is not used as it cannot predict any objects matching the ground-truth for those images. In general, good rank prediction that covers all objects should translate to both high SOR and low MAE simultaneously. Though S4Net has the highest SOR, it also has the highest (worst) MAE. It means that S4Net only performs well to predict a small subset but not all salient objects and their ranks. SOR excludes any missing objects and does not penalise such missing prediction. The high MAE of S4Net indicates both incorrect prediction of saliency maps and object ranks.

CPD-R produces the best MAE score. However, the saliency maps produced are usually not as smooth as ours, and non-salient areas often filled with false saliency values. This can be seen in Figure 5.10, which shows CPD-R generating saliency maps with missing and weak saliency predictions for parts of salient objects. In the figure, false saliency can clearly be seen in the right-most image from the middle set of examples. The ranking score (SOR) of CPD-R is also inferior to ours. Overall, the proposed method performs the best, with the best SOR using most images while maintaining a low MAE.

Qualitative Evaluation: We showcase results in Figure 5.10 for qualitative comparison. The proposed network directly generates a saliency rank map that segments each object instance

Table 5.2: Ablation study of the proposed model. BbSR refers to the backbone network and the saliency rank network.

Method	MAE ↓	SOR ↑	#Images used ↑
BbSR	0.109	0.773	2353
BbSR+SAM	**0.101**	0.782	**2373**
BbSR+SMM	0.111	0.769	2361
BbSR+SAM+SMM	**0.101**	**0.792**	2365

and predicts their respective ranks simultaneously. The saliency maps obtained from RSDNet [8] often contain many false saliency (e.g. , partial false saliency of paper, third image from the bottom set of examples in Figure 5.10). RSDNet also makes incomplete salient object predictions (e.g. , partial prediction of the person on the right, first image from the top set of examples). S4Net [17] often predicts wrong and fewer object proposals than ours. Fewer object proposals lead to less available objects for SOR calculation and thus unreliable SOR score. BASNet [18] produces cleaner results. However, BASNet, RSDNet, CPD-R [19] and SCRN [20] often mix up the respective object ranks. This validates the effectiveness of our saliency rank approach that infers attention shift order.

5.4.2 Ablation Study

We perform an ablation study to evaluate each of the proposed components, in Table 5.2. From the table, we can see that adding SAM to the base model improves performance across all metrics. However, adding SMM does not help improve performance. Considering that the output features from SMM are 1-D feature vectors, we lose some spatial information that could be useful for boosting segmentation quality and therefore, the performance of MAE. Additionally, the features do not directly interact with the object segmentation features and so further segmentation quality is not improved. The table shows that SMM mainly enhances the ranking performance (SOR) when combined with SAM. This suggests that SMM learns to correlate the spatial information of salient objects with the image feature, and increase saliency ranking performance. The full model has the best overall performance. It provides the highest SOR score using large number of images. The MAE is also tied as best. These show the effectiveness of the proposed components.

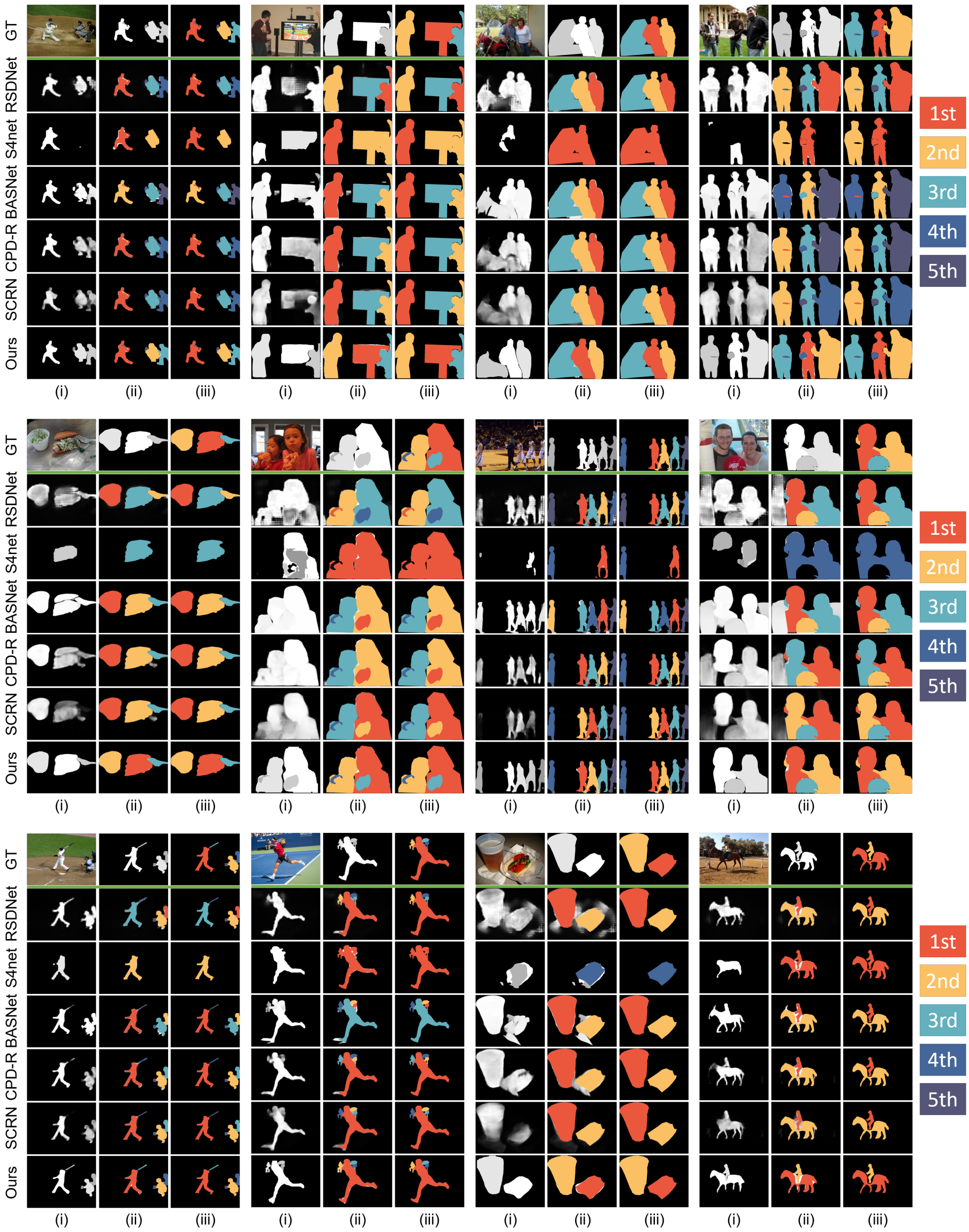


Figure 5.10: Comparison of the proposed method with state-of-the-art methods: RSDNet [8], S4Net [17], BASNet [18], CPD-R [19] and SCRN [20]. Each example in the top row shows the input image, ground-truth saliency map and ground-truth ranks, while for the following rows: (i) saliency prediction map, (ii) saliency prediction map with predicted rank of ground-truth object segments colourised on top, and (iii) corresponding map that contains only the predicted rank of ground-truth objects. The result in (iii) is leveraged to obtain the predicted saliency ranks for quantitative evaluation.

5.4.3 Further Comparison with S4Net

Table 5.3: Quantitative comparison with S4Net for the task of salient instance detection on our dataset. Note that we do not include comparison with RSDNet, BASNet, CPD-R and SCRN since they are unable to perform this task.

Method	mAP^r@0.5 ↑	mAP^r@0.7 ↑
S4Net [17]	16.9 %	10.7 %
Ours	**57.4 %**	**48.3 %**

Like S4Net [17], our network is able to generate individual segmentation for each salient object instance. We further compare our network to S4Net on the task of salient instance detection. We do not include comparison with RSDNet [8], BASNet [18], CPD-R [19] and SCRN [20] as they are unable to produce output of salient object instances. We use the mean Average Precision (mAP^r, $r = 0.5/0.7$) to measure the performance similarly as in [17]. Table 5.3 reports the results between S4Net and our network for salient instance detection on our dataset. The table shows that our network outperforms S4Net by a large margin. The results reveal that S4Net is not able to handle the primary task of salient object ranking, which is the focus of this chapter. S4Net predicts very few salient objects when compared to our network (see Figure 5.10) and misses the prediction of saliency towards corresponding ground-truth objects in over one third of the test set (indicated by #Images used in Table 5.1).

5.4.4 Saliency Ranking on Different Contexts

Our study proposes the first deep network to model human attention shift. Our approach is based on bottom-up and top-down inference, which aligns closely to human visual processing. In the design, we have not fully explored scene context (we have only used spatial context and global image features), yet the results is promising. Spatial context correspond to the size and spatial location of objects in relation to the image scene. The global image context features correspond to prominent features in the image, which establish the scene setting.

Our network learns to reason the saliency rank of individual object features against the global features of an image scene. Such learning can also capture relationships between separate image features and corresponding saliency ranked objects. Figure 5.11 showcases examples of different image scenes containing “sports ball”. We select this object category for this investigation as it can be found in diverse scenarios. It is generally salient and semantically meaningful to the scene context in most of the images it is involved in. The figure demonstrates

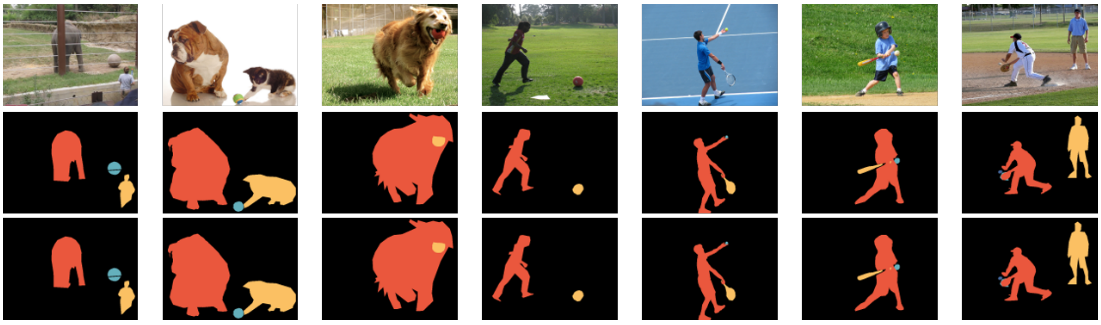


Figure 5.11: Example scenes containing “sports ball” object category. Images from our dataset (Top row), GT Ranks (Middle row), our network rank prediction (Last row).

that our network is able to learn relationships between the object category and various image scenes, while correctly rank the object categories.

5.5 Limitations and Future Work

A limitation of our current method is that it is performed in two stages. In the first stage we pre-generate backbone and object features. Then, we use the pre-computed features as input for training the rest of the saliency rank network. Our current network is not end-to-end. During saliency rank training we are unable to fine-tune all the layers (including the backbone), therefore, training is not at optimum. The main reason we perform our training in two stages is due to our network exceeding the memory limit in the GTX 1080Ti GPU. There are two solutions for this issue, where one is to simply use a more powerful GPU with larger memory capacity. The second is to efficiently reduce the number of training parameters in the network, which will be quite a difficult task. Adopting a light-weight backbone network is a possible approach, but this will usually come with a cost of decreased performance.

Another weakness with our current method is that we perform ranking by predicting rank classes for each object. We observe that multiple objects may be given the same rank class. We resolve this by using the predicted rank classes and corresponding probabilities in a scoring mechanism during post-process. However, this is not optimal as the network will not be able to learn the rank differences between multiple objects efficiently.

While we have observed limitations with our proposed method in this chapter, we will aim to address these issues in the next Chapter 6.

5.6 Summary

In this chapter, we proposed to our knowledge the first saliency rank dataset based on attention shift order. The dataset is motivated by psychological studies and behavioural observations, and is supported by our user study, that humans attend salient objects one at a time and in an order of decreasing values of saliency. We also proposed a novel bottom-up and top-down saliency rank prediction approach that infers attention shift order. The proposed approach performs favourably against several state-of-the-art methods on the proposed saliency rank dataset.

Chapter 6

Saliency Ranking with Learning to Rank Supervision

Contents

6.1 Introduction **52**

6.2 Proposed Method **54**

6.2.1 Network Architecture Overview 54

6.2.2 Salient Instance Edge Module (SIEM) 55

6.2.3 Saliency Rank Network 56

6.3 Experiments **56**

6.3.1 Comparison with State-of-the-Arts 57

6.3.2 Evaluation on Additional Data 59

6.3.3 Evaluation on Salient Instance Segmentation 61

6.3.4 Ablation Study 62

6.3.5 Saliency Ranking on Different Contexts 64

6.4 Limitations and Future Work **64**

6.5 Summary **64**

6.1 Introduction

During the publication of the work in the previous Chapter 5, the authors of [8] have since extended their work and propose a new COCO-SalRank dataset [89]. Unlike the rank modified PASCAL-S dataset, their new dataset does not contain tied saliency ranks. However, their ground-truth rank generation uses hand-designed criteria for producing fixation maps, which are then applied to determine instance ranks. Similarly, Liu *et al.* [2] follow [89] for the relative salient object ranking task. They also propose a new saliency ranking dataset by utilising the fixation maps from the SALICON [105] dataset with MS-COCO [27]. Both [89] and [2] model relative saliency ranking based on fixation maps that are generated from applying a Gaussian filter onto fixation data, which however does not consider the process of shifting attention that is performed by the human visual system. To re-iterate, our previous work and the work in this chapter focuses on salient object ranking based on attention shift.

Recently, [21] follows our study of saliency ranking based on attention shift. They propose a Position-Preserved Attention module to incorporate positional information with object features to improve saliency ranking. We also propose a Spatial Mask Module (see 5.3.4) to capture object positional information and concatenate with object features. The main difference between ours and [21] is how we extract and embed our positional information. Our module allows explicit learning of the relationship between object position and scene for saliency ranking. In our experiments (Section 6.3) our method outperforms existing techniques and achieves new state-of-the-art results.

As mentioned in Chapter 5, our rank network is not trained in an end-to-end manner, but in two stages. We pre-generated our backbone and object features during stage 1 and fed them as input during stage 2 of saliency rank training. This is not ideal as it meant that our backbone and object features are fixed and unable to fine-tune with saliency rank training. We resolve this by modifying our previous saliency rank network, so that it can train end-to-end in this work.

Another limitation of our initial saliency rank approach is that it considered ranking as a rank-id classification task, which can lead to tied rank predictions. Objects are predicted a particular rank class and it is combined with their classification probabilities for rank scores. Descending rank scores would then provide the final saliency ranks. This scoring system is performed during post-process, and therefore not an optimal solution for learning to rank. The network does not have the ability to learn the rank scores and differences between multiple objects. The work in this chapter tackles this problem by transforming the rank network to

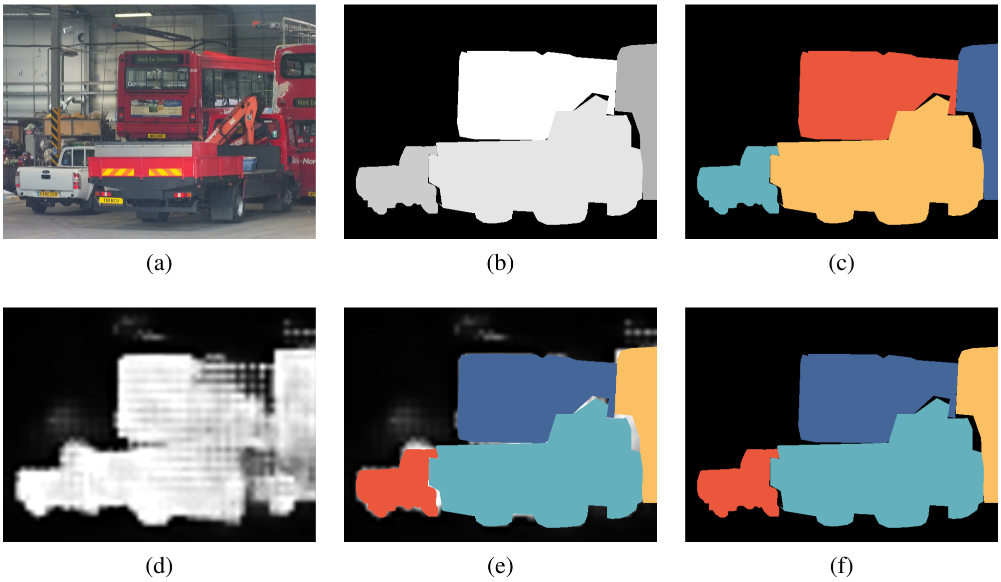


Figure 6.1: (a) Example input image, (b) corresponding ground-truth (GT) saliency rank, (c) corresponding GT saliency rank (colourised), (d) saliency rank prediction by RSDNet [8], (e) corresponding saliency rank by RSDNet (GT objects overlaid and colourised) and (f) corresponding saliency rank by RSDNet with only GT objects (overlaid and colourised). Note that the accuracy of predicting correct ranks can be impacted by the quality of the objects captured.

predict rank scores, while trained with a list-wise ranking loss.

We further expand on the issue of saliency ranking impacted by the quality of objects being captured. As shown in Figure 6.1, there is no clear boundaries and distinction between the objects, which causes the rank prediction to be completely incorrect. In this work, we focus on enhancing the details around the boundaries of an object by introducing an edge segmentation module, which improves the overall segmentation of objects and the final saliency rank prediction.

Contributions: In this chapter, we extend the work from the previous Chapter 5 based on the limitations we have observed. The main contributions of this work include:

- We propose a new Salient Instance Edge Module (SIEM) and pair it with the instance mask segmentation branch in order to mutually improve the segmentation of salient instance masks and edges. This further boosts the performance of saliency ranking, as it enables the network to distinguish salient instances from the background and other objects better.

- We make significant modifications to our previous salency rank architecture in Chapter

5, including listwise ranking loss for saliency ranking instead of formulating it as rank-id classification, top-12 object proposals for salient object and rank reasoning, and end-to-end training with warm-up and fine-tuning.

- We further experiment and evaluate the proposed method, including additional comparisons with state-of-the-arts, ablation study, introducing a new metric for saliency ranking, and evaluation on an additional dataset.

The work in this chapter has been submitted to[1].

The following sections explain the modified and extended saliency rank method in Section 6.2, show our experimental results in Section 6.3, discuss limitations in Section 6.4 and conclude the chapter in Section 6.5.

6.2 Proposed Method

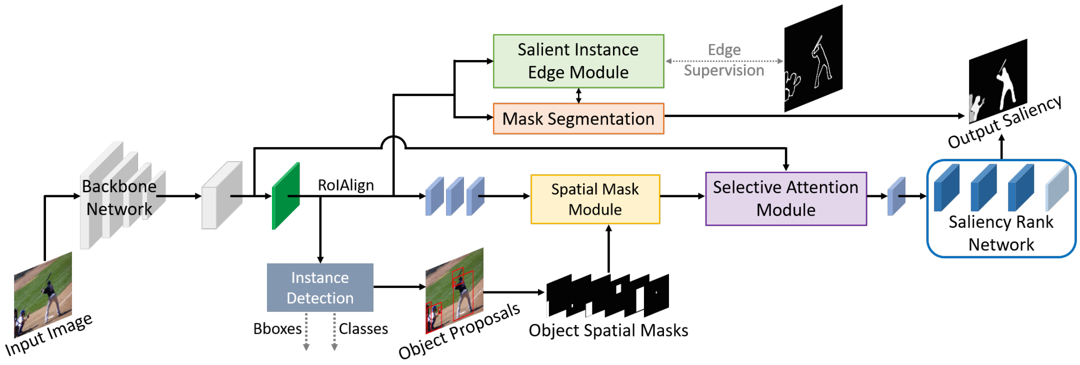


Figure 6.2: Architecture Overview. In this work, we modify our previous saliency rank network from Chapter 5. Specifically, we enable end-to-end training, adjust network parameters, introduce a new Salient Instance Edge Module (SIEM) and transition from rank-id classification to rank score prediction.

6.2.1 Network Architecture Overview

We build on top of our original saliency rank network from Section 5.3. We first modify our network so that it can perform end-to-end training with a warm-up and fine-tuning training strategy. We adjust the parameters of our network to generate the top-12 object proposals. This improves training by reducing the number of distractors from learning the top-5 saliency rank

[1] Avishek Siris, Jianbo Jiao, Gary K.L. Tam, Xianghua Xie and Rynson W.H. Lau. "Inferring Attention Shifts for Salient Instance Ranking". *International Journal of Computer Vision (IJCV)*, 2022.

objects. Next, we introduce a new Salient Instance Edge Module (SIEM). It is coupled with the segmentation branch from our backbone to jointly enhance prediction of salient instance segmentation and edges. The pairing of instance segmentation and edges cooperatively boost their prediction. Furthermore, the joint supervision helps the network distinguish instances and their saliency rank from other objects and background. Finally, we transform our small ranking network so that it predicts rank scores instead of rank-id classes.

6.2.2 Salient Instance Edge Module (SIEM)

Our dataset contains images with many objects and noisy background features. In these images, salient objects may be very close to other objects and may have features similar to the background. This introduces noise to the boundaries of salient objects and can make it difficult to distinguish salient objects from other objects or the background. As a result, accurately segmenting and predicting salient ranks of multiple salient instances can be challenging. This can be seen in Figure 6.1(d-f), where traditional methods do not explicitly capture individual objects and can have difficulties in differentiating between multiple objects. Based on these observations, we propose the Salient Instance Edge Module (SIEM), and jointly train it with the mask segmentation branch. The coupling of the instance mask and SIEM refines their predictions mutually. This further propagates enhancement to the salient instance segmentation and ranking tasks.

Both the mask segmentation and SIEM have the same structure, but the connection between the networks differ. Figure 6.3 illustrates the structures of SIEM and mask segmentation branch. Mask segmentation first contains four convolutional layers. We then add a fusion layer to combine the mask features with the edge features by addition. The final mask segmentation is then generated with a final convolutional layer and a prediction layer. Likewise, SIEM contains four convolutional layers followed by a fusion layer, and final convolutional and prediction layers. Here, the fusion layer differs as we subtract the edge features with the mask segmentation features. The resulting features are then concatenated with a residual connection. The two fusion layers enable the networks to use features from the other network to effectively focus on their particular task. For example, the addition of edge features to the mask features (M-Fuse in Figure 6.3) allows the mask segmentation network to accurately capture the shape of instances, while the subtraction in the SIEM (E-Fuse) forces the network to focus around regions of the mask. We adopt the boundary loss as in [127] for training our edge module.

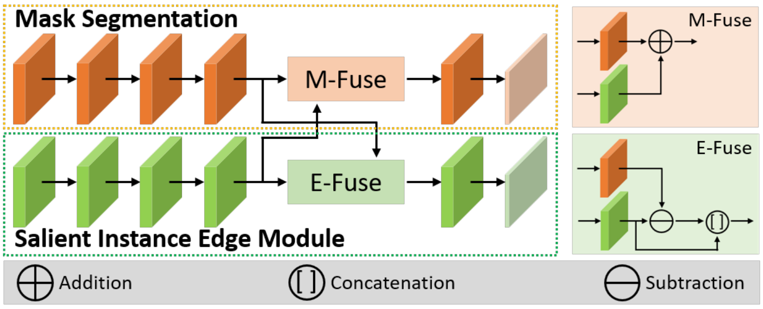


Figure 6.3: Details of the Salient Instance Edge Module (SIEM) and Mask Segmentation. Orange Blocks: mask segmentation features. Green Blocks: edge features. The orange and green blocks represent features obtained after applying convolutional layers. Both mask segmentation and edge features are of shape (14 × 14) with 256-D feature channels. Transposed convolutional layers (5th blocks) increases the mask segmentation and edge features to size (28 × 28). The final orange and green blocks (with lighter shade) represent the mask segmentation and edge predictions for individual objects.

6.2.3 Saliency Rank Network

We transform our rank network from rank-id classification to rank score prediction. We employ a simple ranking network to predict rank scores for salient instances. Our rank network consists of three fully connected layers and a final scoring layer. Inspired by the learning to rank problem [128], we adopt the list-wise loss, ListMLE [129], as our ranking loss.

During inference, we combine the saliency rank scores with object segmentation (from the mask segmentation branch) to generate the final salient object rank map. Like [8], which determines object saliency rank by the descending average pixel saliency value of each object. We consider the top-5 available salient objects based on their descending score values and assign discrete ranks. The salient object rank map reveals the ranks of each salient object by displaying unique grey-scale values (based on their rank) for their segmentation.

6.3 Experiments

We follow similar experimental settings as in Section 5.4, but with changes according to the modifications we have introduced in this work.

Our new model is built on top of the previous version with Tensorflow and trained on an RTX 3090 GPU. During saliency rank training, we use a warm-up strategy by freezing the backbone layers and training the rest. We then fine-tune all the layers together. We set the mini-batch size to 8 during warm-up and 2/4 for fine-tuning depending on memory limitations. We train variations of the network up to 30 epochs for warm-up and 10 epochs for fine-tuning.

We use the SGD optimizer with gradient norm clipping set to 5. The learning rate is set to 10^{-3} for warm-up. For fine-tuning, we set the learning rate to 10^{-8} for the backbone layers and 10^{-6} for the rest. Momentum and weight decay are configured as 0.9 and 10^{-4}, respectively.

We use our proposed saliency rank dataset from the previous chapter (Section 5.2) for testing. Again, we use the Salient Object Ranking (SOR) and MAE metrics for evaluation. Additionally, we adopt the nDCG (normalised Discounted Cumulative Gain) [130] to measure the consistency between the predicted and GT ranks. We define the metric as:

$$nDCG = \frac{DCG}{Ideal(DCG)}, \tag{6.1}$$

$$DCG = \sum_{i=1}^{k} \frac{rel_i}{\log_2(i+1)}, \tag{6.2}$$

$$rel_i = 5 - abs(r_{GT} - r_p), \tag{6.3}$$

where rel_i is the relevance score of the i^{th} object. k is the maximum number of GT objects per image. r_{GT} and r_p are the GT rank and predicted rank of object i. $Ideal(DCG)$ is the best score when the rank order is perfect, which is a relevance score of 5 for each object.

6.3.1 Comparison with State-of-the-Arts

Quantitative Evaluation: We compare against three ranking methods: RSDNet [8], IL-RSR [2] and SOR-PPA [21]. We also compare with thirteen state-of-the-art salient object detection methods: S4Net [17], BASNet [18], CPD-R [19], SCRN [20], PFANet [25], EGNet [22], ITSD [26], MINet [24], LDF [23], CSNet [28], GateNet [29], VST [30] and SCAS [3]. Note that these methods (except S4Net and SCAS) do not predict individual object segmentation and instead only provide a single binary saliency map.

S4Net and SCAS have a similar structure to our backbone and outputs object instance segmentation. We modify both S4Net and SCAS in order to predict up to 6 classes (5 Ranks + 1 BG) for each object instead of the binary prediction as in their original papers [3, 17] for a fair comparison. We then use the predicted rank classification and descending score probabilities to obtain distinct saliency ranks. For the rest of the salient object detection models and RSDNet, the predicted saliency ranks of ground-truth objects are obtained by averaging the pixel saliency values. The rank order of top-5 available objects are selected by descending order of such averages, which are then converted to discrete ranks.

We clarify that RSDNet is directly evaluated on our dataset using its pre-trained weights. When we try to adapt and train their model on our dataset (using their available source code),

Table 6.1: Comparison with the state-of-the-art methods on our dataset (Section 5.2). The top section consists of salient object detection methods and middle section contains saliency ranking methods. The bottom section contains our methods, with ASSR [1] being our earlier model from Chapter 5. Note that RSDNet scores are based on direct prediction with pre-trained weights from their datasets. #Images used refers to the number of predicted images usable for SOR calculation. ↑ (↓) means the higher (lower) the better. Overall best performance is shown in **bold**.

Method	MAE↓	SOR↑	#Images used↑	nDCG↑
S4Net [17]	0.150	0.681	1661	0.674
BASNet [18]	0.115	0.696	2321	0.787
CPD-R [19]	0.100	0.763	2394	0.837
SCRN [20]	0.116	0.756	2416	**0.843**
PFANet [25]	0.156	0.741	**2418**	0.834
EGNet [22]	0.097	0.764	2413	0.842
ITSD [26]	0.098	0.729	2416	0.823
MINet [24]	0.099	0.706	2415	0.814
LDF [23]	0.093	0.734	2413	0.828
CSNet [28]	0.136	0.738	**2418**	0.831
GateNet [29]	0.094	0.719	2417	0.820
VST [30]	0.093	0.766	2411	0.841
SCAS [3]	0.100	0.717	2118	0.758
RSDNet [8]	0.139	0.728	**2418**	0.827
IL-RSR [2]	**0.091**	0.726	2239	0.785
SOR-PPA [21]	0.098	0.755	2351	0.801
ASSR [1]	0.101	0.764	2333	0.822
Ours	**0.091**	**0.800**	2371	**0.843**

the model does not converge. We thus use their model with the pre-trained weights to evaluate on our dataset.

Table 6.1 shows the experimental results. It shows that our method outperforms other methods on the proposed dataset, achieving the best overall performance with better scores among all performance measurements (MAE, SOR and nDCG). Note that RSDNet, PFANet and CSNet use all images during the SOR calculation, as their single binary saliency maps often contain many false saliency. Noise or very weak saliency is often propagated throughout the image and reach parts of objects. This allows RSDNet, PFANet and CSNet to obtain saliency rank by averaging object pixel values to cover most objects. However, their MAE and ranking performance is near the lower end.

We have the best MAE performance tied with IL-RSR, although IL-RSR has much lower scores for SOR and nDCG rank metrics. One reason for this is that IL-RSR is able to capture the overall segmentation of most GT objects, resulting in a good MAE score, but fail to predict the rank order between objects accurately, resulting in lower rank scores. Our extended method proposed in this work has gained a significant improvement on the SOR rank metric from our initial work (ASSR, Chapter 5). It produces the best results with a 4.43 % performance gain over the second best method. SCRN has the highest nDCG score also tied with our method, as nDCG awards higher scores for correct predictions of top ranks. Nonetheless, SCRN has much lower MAE and and SOR scores than ours. This suggests that SCRN is mainly able to predict the top ranks well, but is unable to correctly predict the lower ranks.

In general, good rank predictions should translate into both high SOR and nDCG scores but low MAE score simultaneously.

Qualitative Evaluation: Figure 6.4 showcases qualitative comparison results. We compare our method with state-of-the-art methods that are specifically designed for saliency ranking. We can see that the saliency maps obtained from RSDNet [8] often do not capture all the GT objects well, which can lead to incorrect rank predictions. Even if an instance is captured well by the instance detection method, it is still difficult to correctly rank the GT objects. In contrast, our method is able to segment the overall shape of most GT objects and correctly rank them. It is also able to rank multiple objects in complex image scenarios, where there are cluttered and similar visual features among multiple objects (e.g. , fourth and fifth columns).

6.3.2 Evaluation on Additional Data

In addition to the experiments on our dataset, we also evaluate on the relative saliency ranking dataset from [2]. Note that this dataset is not based on attention shift, whereas our saliency rank dataset is defined by attention shift (Section 5.2). They define saliency rank based on the order of maximum saliency intensity value for each object from a fixation map. The fixation map does not consider the sequential order of fixations, but rather simply encapsulates the density of fixation points around objects. This does not follow the idea of saliency ranking from the order of sequential shift between multiple objects. The differences in the GT rank generation process between their dataset and ours is demonstrated in Figure 6.5. We test our method on their dataset for generalisability. Note that we do not test on the dataset from [89] as it is not released.

For this experiment, we modify both our network and SOR-PPA [21] to train and predict up to 8 GT saliency ranks. Table 6.2 compares between ours and two instance-based state-of-the-arts designed for saliency rank prediction. Note that the evaluation results are based on 2787

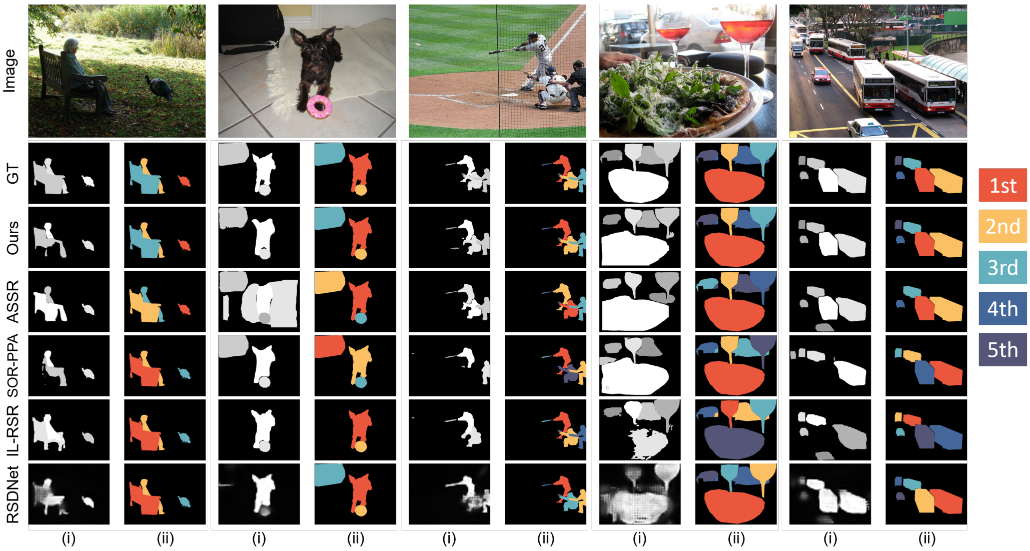


Figure 6.4: Comparison of the proposed method against state-of-the-art methods designed for saliency ranking: ASSR [1], SOR-PPA [21], IL-RSR [2] and RSDNet [8]. Each example in the first row shows the input image, while the second row show ground-truth saliency map (i) and ground-truth saliency ranks (ii). The following rows are predictions from the compared methods, where (i) saliency prediction map, (ii) corresponding map that contains only the predicted rank of ground-truth objects. The result in (ii) is leveraged to obtain the predicted saliency ranks for quantitative evaluation.

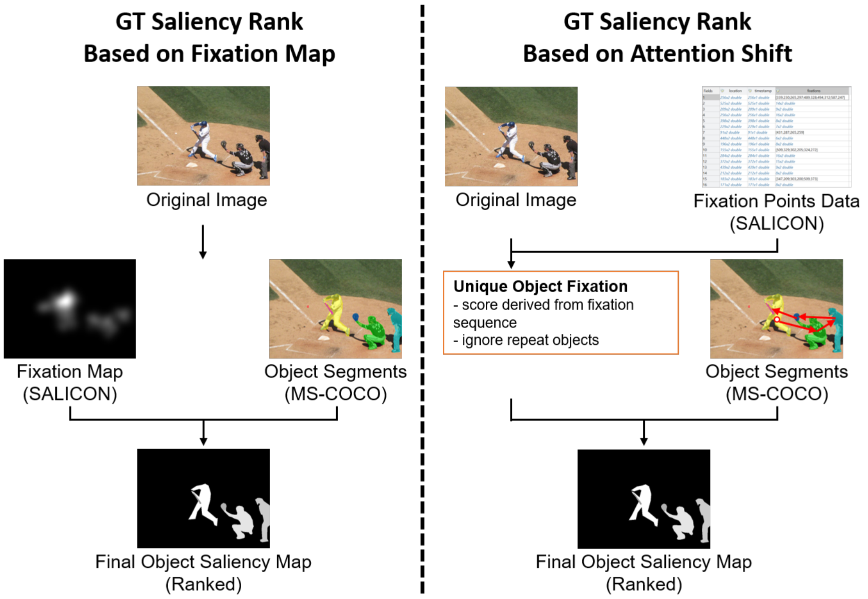


Figure 6.5: Simple illustration of the GT saliency rank generation process between the dataset in [2] and ours (Section 5.2). [2] uses the max intensity value inside the segmentation of objects from the fixation map to assign saliency rank in descending order (left). Our GT rank order is based on unique attention shifts between objects using the sequential fixation data (right).

Table 6.2: Comparison with instance-based state-of-the-art methods on the dataset from [2] for testing generalisability. #Images used refers to the number of predicted images usable for SOR calculation. ↑ (↓) means the higher (lower) the better. Best scores are in red, while second best scores are in blue.

Method	MAE↓	SOR↑	#Images used↑	nDCG ↑
IL-RSR [2]	0.110	0.782	2695	0.903
SOR-PPA [21]	0.119	0.532	2744	0.859
Ours	0.099	0.739	2759	0.915

test images from the original 2929 test images. We find that the dataset in [2] contains 142 test images, where the rank data consist of multiple instances with the same rank and/or more than the 8 max GT ranks. Table 6.2 shows that our method achieves the best MAE and nDCG scores and second best on the SOR score. This indicates that our method can predict most of the ground-truth salient objects well. It is also able to predict correct ranks for top-ranked objects more consistently, and only experiences some difficulties for ordering lower-ranked objects. Overall, our method generalises well to the dataset from [2], even though the dataset is not built on our definition of attention shift for saliency ranking. Given that our method shows strong results on MAE and nDCG, we believe our method can improve further (especially on SOR) with adjustments to network configuration and training parameters.

6.3.3 Evaluation on Salient Instance Segmentation

Table 6.3: Quantitative comparison with S4Net for the salient instance segmentation task on our dataset. Note that we do not compare with other state-of-the-arts since they are unable to perform this task.

Method	mAP^r@0.5 (%) ↑	mAP^r@0.7 (%) ↑
S4Net [17]	16.7	10.6
IL-RSR [2]	48.2	38.3
SCAS [3]	38.6	27.6
SOR-PPA [21]	55.1	47.1
ASSR [1]	60.6	51.0
Ours	**64.4**	**53.8**

We compare our network, IL-RSR [2], SCAS [3], SOR-PPA [21] and ASSR [1] with S4Net [17] on the salient instance segmentation task. We omit the other state-of-the-art methods from this experiment, as they are unable to produce salient object instances. We use mean Average Precision (mAP^r, $r = 0.5/0.7$) to measure the performance, as in [17]. Table 6.3 reports the results on our dataset. Our network outperforms S4Net and other state-of-the-arts

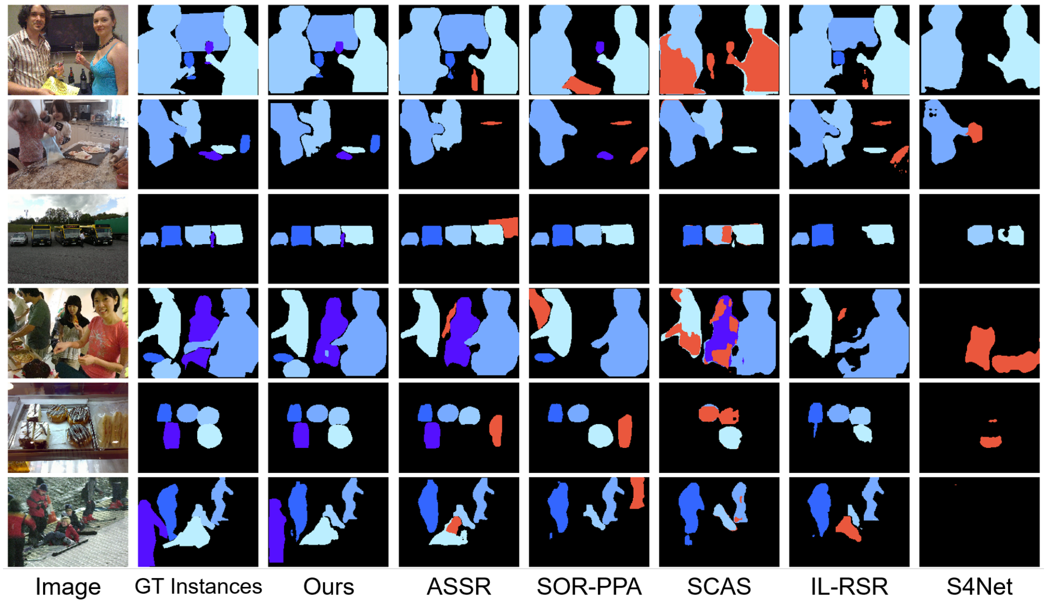


Figure 6.6: Qualitative comparison of instance-based salient object segmentation. Instances with 5 different shades of blue are predicted instances that match with their corresponding GT instance. Red instances represent false predictions.

by considerable margins. Figure 6.6 shows that our network can predict multiple instances with good segmentation accuracy. It can capture smaller objects that are difficult to distinguish from the surrounding. Other state-of-art methods tend to introduce false salient instances and fail to segment all GT instances. Our extended network also improves upon our previous work (ASSR) and achieves the best performance.

6.3.4 Ablation Study

We update the ablation study with the new modifications in this work. Table 6.4 shows the full model with the best overall performance across all metrics. It produces the highest SOR and nDCG scores. MAE is also tied best. Addition of each component to the base model of the proposed method generally improves the performance across the metrics.

When comparing with the previous ablation study (Section 5.2), the transformation of our rank network from rank-id classification to rank score prediction has increased results of the base model. We see a more substantial gain in SOR score when we add SIEM. This suggests that explicitly using edge information enables the network to improve object segmentation, consequently helping differentiate between close objects and boost rank order prediction.

Table 6.4: Ablation study of the proposed model. BbSR refers to the backbone network plus the small saliency rank network.

Method	MAE ↓	SOR ↑	#Images used ↑	nDCG ↑
BbSR	0.096	0.787	2373	0.835
BbSR+SMM	0.092	0.793	**2377**	0.842
BbSR+SAM	0.093	0.788	2366	0.839
BbSR+SIEM	0.092	0.798	2372	0.842
BbSR+SMM+SAM	0.092	0.794	2366	0.842
BbSR+SMM+SIEM	**0.091**	0.799	2373	0.842
BbSR+SAM+SIEM	0.092	0.795	2371	0.841
BbSR+SMM+SAM+SIEM	**0.091**	**0.800**	2371	**0.843**

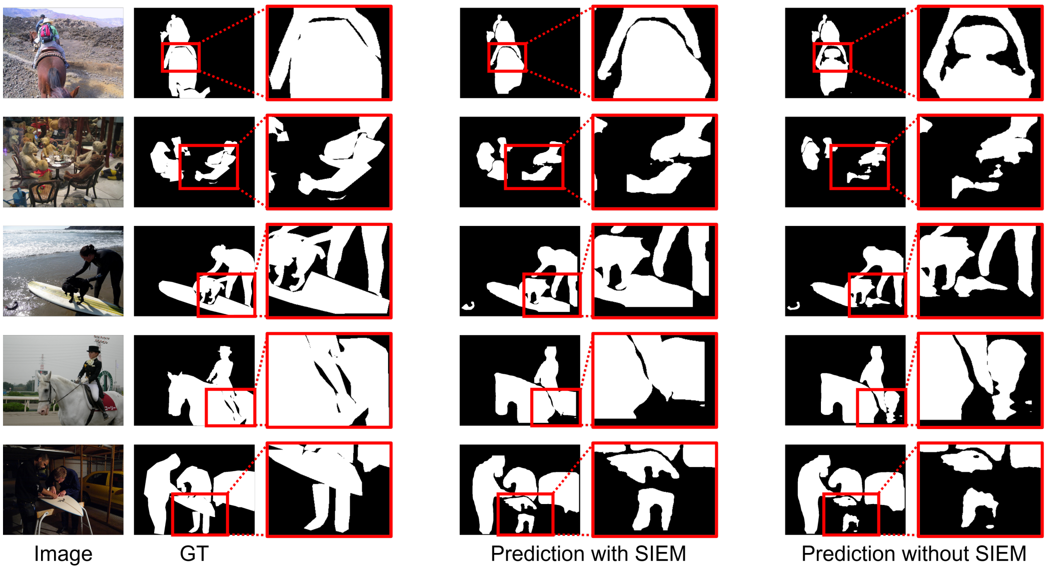


Figure 6.7: Qualitative comparison of salient object segmentation between our full model with Salient Instance Edge Module (SIEM) versus full model without SIEM.

Figure 6.7 compares the salient object segmentation accuracy between our full model with the Salient Instance Edge Module against the full model without the edge module. It shows that the proposed edge module coupled with mask segmentation enhances the capture of salient instances. The top example in the figure shows SIEM helps improve the overall body segmentation of the horse, while clearly separating it from the person riding the horse. These results show the effectiveness of the proposed components.

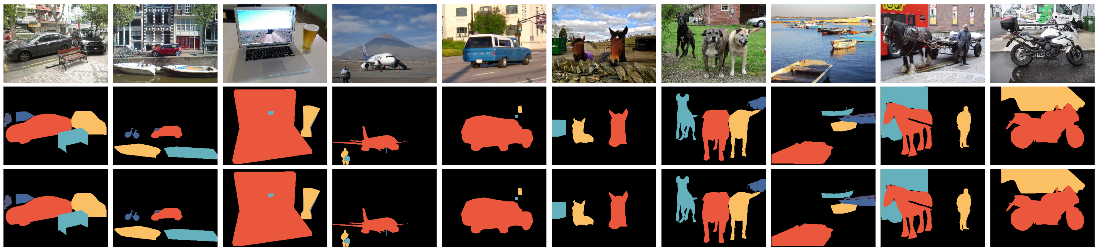


Figure 6.8: Example scenes containing objects in the *vehicle* category. Input images from our dataset (Top row), GT Ranks (Middle row), rank prediction from our method (Last row).

6.3.5 Saliency Ranking on Different Contexts

We further investigate the performance of our saliency rank method on different scene context. Figure 6.8 showcases examples of different image scenes containing "vehicles". The vehicle can be of different sizes, at different locations, or adjacent to other vehicles. Our network learns the relationships between the "vehicle" objects and context (spatial and global image features), and determines the correct saliency rank for each object accordingly.

6.4 Limitations and Future Work

We find that our proposed method can correctly predict the top saliency ranks, but it does experience difficulties in predicting lower ranks, especially on the dataset from [2]. Correctly predicting rank order for lower-ranks is very challenging as the differences among them can be minuscule. A possible future work would be to improve saliency rank prediction for lower ranks.

Our method utilises spatial and global image features as contextual information to improve saliency ranking in different scenes. Nevertheless, we have not fully explored the use of scene context information for saliency. Alternative approaches to extracting semantic scene context could prove valuable for saliency reasoning. In the next Chapter 7, we begin to explore the usefulness of semantic scene context for saliency prediction.

6.5 Summary

This chapter improved upon the proposed method from the Chapter 5 for salient object ranking based on attention shift. Particularly, we modified our initial saliency rank method by enabling end-to-end training with changes to the network parameters. Moreover, we transitioned from rank-id classification to rank score prediction, which increases the overall performance of our method. We also introduced a new Salient Instance Edge Module that enhances the accuracy of object segmentation and subsequently the final saliency rank prediction. We have shown with further comparison against additional state-of-the-art techniques that our method achieves the best results.

Chapter 7

Scene Context Saliency

Contents

7.1 Introduction **68**

7.2 Proposed Dataset **70**

7.2.1 Data Generation 70

7.2.2 Dataset Statistics 73

7.3 Proposed Method **75**

7.3.1 Backbone 75

7.3.2 Semantic Scene Context Refinement (SSCR) 77

7.3.3 Contextual Instance Transformer (CIT) 78

7.3.4 Salient Instance Network 79

7.4 Experiments **81**

7.4.1 Comparison with State-of-the-Arts 81

7.4.2 Comparison on Existing Datasets 83

7.4.3 Ablation Study 85

7.5 Limitations and Future Work **85**

7.6 Summary **87**

7.1 Introduction

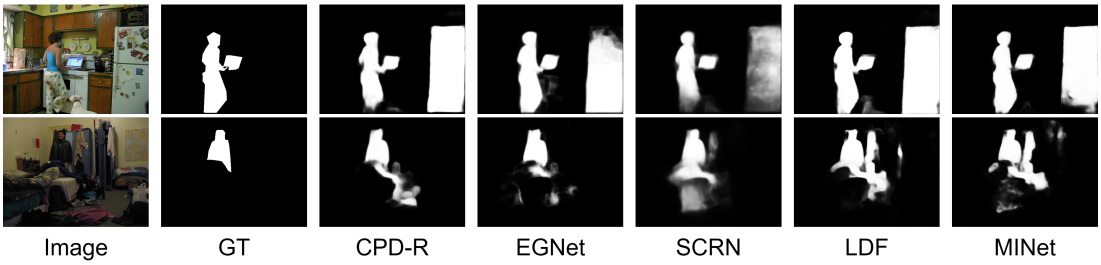


Figure 7.1: Examples of real-world complex scenarios where existing methods (e.g. , CPD-R [19], EGNet [22], SCRN [20], LDF [23] and MINet [24]) may not capture semantic scene contexts well, leading to incorrect detection of distractors.

In this chapter we focus on traditional salient object detection. From our literature review (Chapter 3), we have made observations that lead us to identify problems with existing salient object detection work (Chapter 4). First, many of the common salient object detection datasets are not very challenging. They generally include images with very few objects and simple background, which does not reflect real-world image scenes that contain numerous objects with complex background. This has driven most of the salient object detection techniques to be mainly designed for performing saliency on these datasets. As a result, current state-of-the-art methods struggle to predict accurate saliency on complex images.

In relation, salient object detection models are generally trained only on class-agnostic binary saliency labels. This limits saliency networks in learning semantic contextual information that would help model diverse relationships between objects in a complex scene. Existing techniques do not capture such semantic information and so, they are unable to accurately predict salient objects from distractors. Figure 7.1 shows two examples of real-world complex scenarios where existing models perform poorly. The top row shows a kitchen scene with a salient person and a distractor (e.g. , fridge with similar texture). Existing models can not capture the semantic knowledge of the distractor and are unable to differentiate it from the person's attire, resulting in incorrect saliency for that distractor. It is a similar case for the bottom row which involves a bedroom scene with a salient person surrounded by many distractors (e.g. , clothes and objects with similar texture).

Psychological studies suggest that semantic scene context influences eye movements and attention [131], revealing the relationship between salient objects and the surrounding image scenes. To the best of our knowledge, saliency detection with high-level scene context and spatial context is under-explored, with only two related works [9, 94] addressing a similar problem. Zhang *et al.* [9] propose to leverage captions as the semantic scene context for improving salient object prediction. However, reliance on generated captions can be detrimental to saliency prediction, especially if they are incorrect. On the other hand, [94] derives their scene context features from an image-level scene classification model, whereas the extracted features are too abstract, containing only an overall representation without capturing object relationships within the scene.

The above-mentioned limitations motivate us to explore the use of semantic scene and spatial context for salient object detection in real-world scenarios with complex scenes. To this end, we first construct a novel dataset comprising of images with rich context. We then propose a context-aware saliency modeling framework to leverage semantic scene context features. Specifically, we introduce Instance Context Segmentation and Stuff Context Segmentation to semantically segment *Things* and *Stuff*. These two components perform panoptic segmentation on the whole scene, providing detailed semantics of a given image. However, we find that not all the semantic information play an effective role in defining the semantic scene context of an image. As a result, we propose a novel *Semantic Scene Context Refinement* (SSCR) module to fuse and augment information of salient object features with surrounding semantic scene context for improving saliency reasoning. To further exploit semantic scene context, we propose a *Contextual Instance Transformer* (CIT) to capture the relationship between objects and the scene context.

Contributions: In this chapter, we target salient object detection on real-world complex image scenes. Our main research question is: *Can we learn and use discriminative semantic context to improve saliency modeling in challenging complex scenes with rich context?* We make four contributions in this chapter:

- We propose a semantic scene context-aware framework for salient object detection, which explores the semantic relationship between salient objects and the scene context.
- We propose a *Semantic Scene Context Refinement* module to extract and enhance semantic scene context features that are highly related to the image scene. We further propose a new *Contextual Instance Transformer* to learn the contextual relations between objects and scene context for saliency reasoning.
- We build a new salient object detection dataset with real-world complex scenes to consider semantic scene contexts.

- Extensive experiments demonstrate that the proposed approach outperforms the state-of-the-art methods on our dataset and also generalises well to existing datasets.

This chapter, findings, results, code and data were published in [3][1].

The rest of this chapter is structured as follows: Section 7.2 reveals the process for building our salient object detection dataset consisting of images with complex scenes. We present our proposed method in Section 7.3 and evaluation in Section 7.4. Then we discuss our limitations in Section 7.5 and conclude the chapter in Section 7.6.

7.2 Proposed Dataset

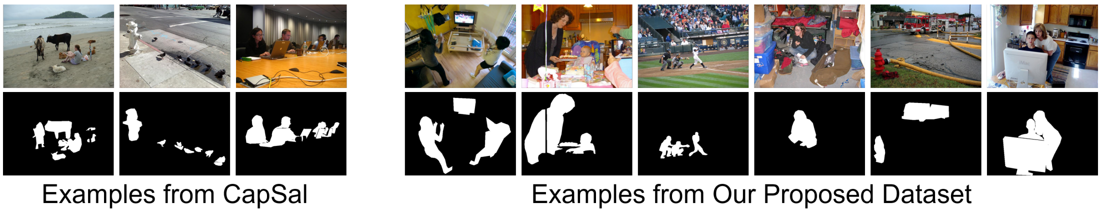


Figure 7.2: Sample images (Top row) and corresponding ground-truth saliency (Bottom row) from the CapSal datatset [9] and our proposed dataset. CapSal contains complex images, but the ground-truth saliency is heavily biased towards the captions data. Whereas, our ground-truth saliency is entirely based on fixation data.

7.2.1 Data Generation

As aforementioned, existing salient object datasets mostly contain images that do not well represent real-world scenes. The CapSal [9] dataset contains real-world images, however the ground-truth salient objects are often heavily biased towards the caption data. The consequence is that all objects that relate to the caption are often considered salient, regardless of whether each of those objects are individually visually salient or not (shown in Figure 7.2).

We build a new dataset to support the modeling of saliency in real-world scenes containing rich semantic context. Our dataset is based on MS-COCO [27] and SALICON [105]. MS-COCO provides images of challenging scenarios and annotations of semantic segmentation of object instances (*Things*) and regions (*Stuff*). SALICON provides the mouse-based fixation

[1] Avishek Siris, Jianbo Jiao, Gary K.L. Tam, Xianghua Xie and Rynson W.H. Lau. "Scene Context-Aware Salient Object Detection". Proceedings of the *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

sequence of respective images. Our dataset is constructed in two phases: 1) automatic ground-truth saliency generation and image filtering, and (2) manual image filtering.

(1) Automatic Phase: We automate the ground-truth salient objects generation based on the observations in [132], where Fosco *et al.* found that humans generally gaze at people during [0,0.5] seconds, and then move towards other objects during [0.5,3] seconds. After the first 3 seconds, there are more fixations on *Stuff* regions. Based on these observations, we collect salient objects if the SALICON fixation points, in the range [0,3] seconds, fall on the MS-COCO annotations of an object segmentation in an image. An object is further labelled as ground-truth salient if more than half of the observers fixate on this object. We automate the process to define ground-truth salient objects on all 15k images using SALICON fixation data.

For each image $I \in \mathbb{R}^{W \times H}$ with spatial dimensions $W \times H$, there are N number of observer fixation data. For each observer $i \in [1,N]$, we augment the fixation sequence to obtain a new fixation sequence F^i. The augmentation includes a) cropping the fixation sequence to at most 3s and b) removing repeated fixations on the same object. We assign a saliency score s_o to an object o, if the j-th fixation $f^i_j \in F^i$ lands on that object.

$$\begin{aligned} s_o &= \sum_i^N \sum_j^T g(f^i_j), \\ g(f^i_j) &= \begin{cases} 1 & \text{if } f^i_j \in P_o, \\ 0 & \text{otherwise}, \end{cases} \end{aligned} \tag{7.1}$$

where T denotes the number of fixations in F^i and P_o refers to the set of pixels belonging to the segmentation of object o. An object is then considered salient in the ground-truth, if its saliency score is greater than half the number of observers for a given image.

Once ground-truth saliency is generated for all available 15,000 images, an automated filtering step is applied to ensure the images are complex and contain rich context. It is a trade-off between complexity of images and the number of resultant images. We find that a minimum of 4 objects and at least 2 object categories per image produce a good set of complex images, whilst retaining a higher number of images in the constructed dataset.

The above automated step however may run into issues when the annotations of foreground and background objects overlap. First, in some of the images in the MS-COCO dataset (e.g. , food on bench / dining table), we observe that the background object (e.g. , bench, dining table) are often large objects. They are incorrectly considered salient simply because the fixations fall on both the foreground and background objects. Figure 7.3 shows examples of such overlap-

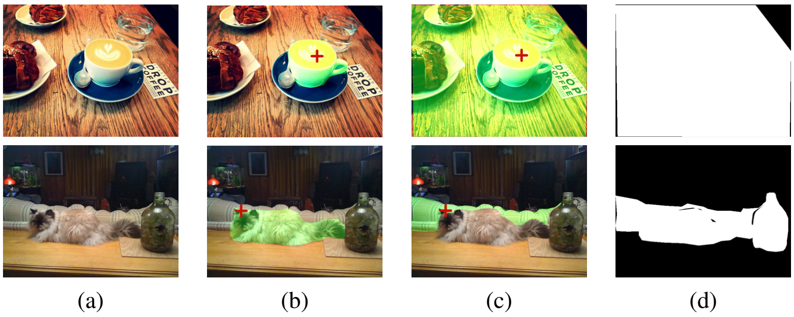


Figure 7.3: Example images of overlapping segmentations that often cause issues in the automatic ground-truth saliency generation. (a) image, (b) foreground object, (c) background object, (d) resulting incorrect saliency map. Red cross in (b) and (c) correspond to an observer's fixation.

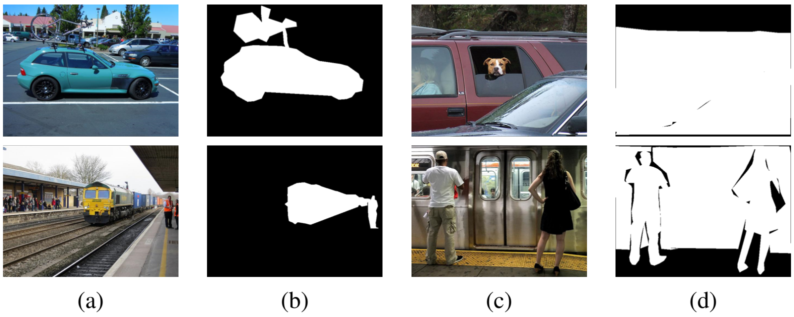


Figure 7.4: Examples of certain object categories (e.g. , car and train) correctly and incorrectly identified as salient in separate scenes, with different object sizes. (a) and (b) are image and corresponding generated saliency map, with those object categories correctly defined as salient, during phase 1 of automatic saliency generation. (c) and (d) are image and resulting saliency map generated with incorrect saliency. The same object categories (car and train) are relatively large and should be considered as background instead.

ping segmentations and the resulting incorrect saliency maps. Second, some objects (e.g. , car, train) are easy to collect fixation as they cover a large portion of the background in some images (e.g. , a person in a car), but they are clearly not salient (e.g. , compared to the foreground person). Figure 7.4 shows examples (e.g. , cars, trains) of such cases and the incorrect saliency maps produced. In the former cases, since MS-COCO does not provide depth information, we manually go through the dataset, identify those background categories and exclude them for saliency ground-truth generation. In the latter cases, we also omit all objects where its area is larger than 60% (a threshold we empirically decided) of the image. These steps are carried out as a pre-filtering step before the above automated process. They ensure that large objects (typically background) are not given saliency scores.

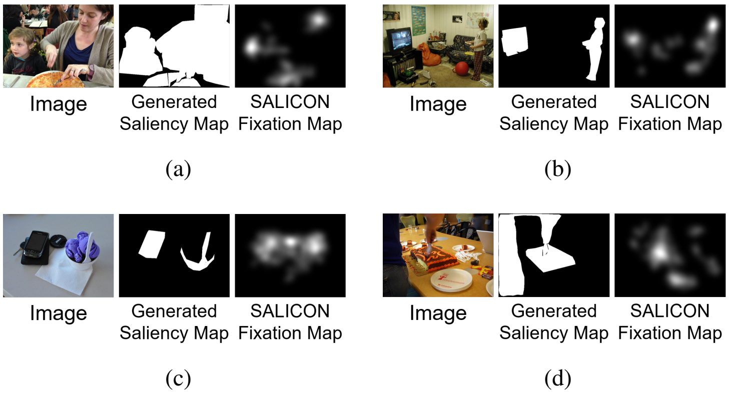


Figure 7.5: Examples of visual comparison during the manual image filtering (Phase – 2) when we construct the proposed dataset. In (a) and (b), the generated saliency maps are comparable to the corresponding SALICON fixation maps, which are kept in our final dataset. Whereas those images with larger discrepancies, as shown in (c) and (d), are removed from the dataset.

(2) Manual Phase: To ensure the quality of the constructed dataset, we manually inspect if the generated salient object map from phase-(1) is consistent with the corresponding SALICON fixation map, following a similar procedure in [9]. Specifically, we would like to ensure that the peak fixations in SALICON also land on objects that are identified as salient in our generated saliency maps. For example, Figure 7.5(c) and 7.5(d) show two images that are removed, because there are large discrepancies between the peaks of fixation maps and the chosen salient objects in the generated saliency maps. This step removes inconsistent annotations that may arise from the automatic process.

After the two phases, our final dataset consists of 5,534 training and 2,554 testing images.

7.2.2 Dataset Statistics

Table 7.1 compares the average number of objects and categories per image in existing datasets and our dataset. Existing datasets do not provide object segmentation or category data. Therefore, we report the statistics in the table by sampling 25 images randomly from each dataset and manually counting the objects and categories. It shows that our dataset contains images with a higher count of objects and categories that is much closer to real-world scenes.

Figure 7.6 shows the distribution of object and Stuff region categories in the proposed dataset. We can see that the "person" category is most prevalent throughout the dataset, which is quite expected as photos are commonly taken with people as one of the main targets. In terms of stuff region categories, there is a balance and it is not dominated by a single category.

Table 7.1: Comparison of the average number of objects and object categories per image among existing datasets and our dataset.

Dataset	#Avg. Obj.	#Avg. Obj. Cat.
ECSSD [10]	1.32	1.28
PASCAL-S [11]	2.08	1.80
HKU-IS [12]	2.12	1.68
DUT-OMRON [13]	1.44	1.24
DUTS [14]	1.56	1.36
Ours	12.79	4.62

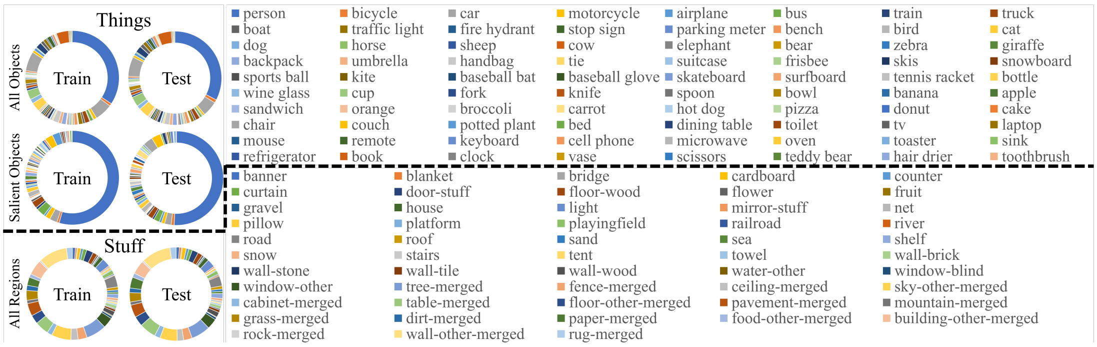
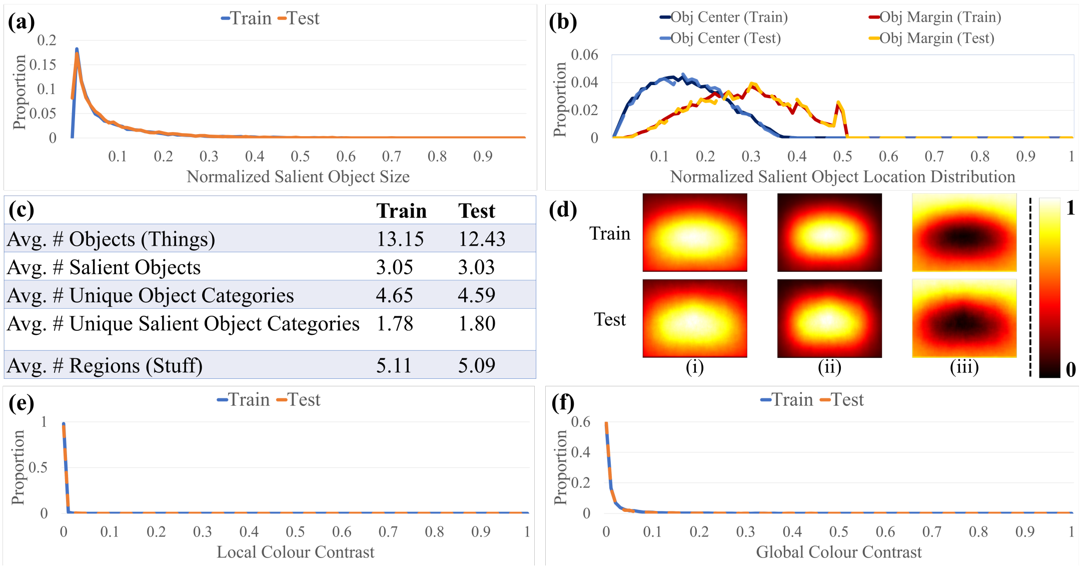


Figure 7.6: Statistics of the proposed dataset, presenting the distribution of all objects (*Things*) and *Stuff* region categories in the dataset.

The distribution of the sizes of salient object shown in Figure 7.7 (a) indicates that our salient objects are generally of smaller scale with respect to the image size. Figure 7.7 (b) displays the distribution of distances between salient objects and the image centre. Figure 7.7 (c) reports the statistics of the objects, object categories and stuff regions. The statistics show that our dataset contains images with complex image scenes. Figure 7.7(d) visualizes the overlay map from locations of all objects (i), salient objects (ii) and Stuff regions (iii), respectively in an intensity map. We also report the contrast of foreground salient objects and surrounding background in local (e) and global (f) views (following [11]). The local contrast compares the colour contrast between foreground (salient object) and background in the local boundary of each salient object. Conversely, global contrast compares the colour contrast of foreground (salient object) and background from the entire image for all salient objects. These graphs show that our GT salient objects have lower colour contrast to their surroundings. This makes the GT salient objects more challenging to detect. It suggests that top-down factors may be more useful for our dataset, while simple low-level contrast is unlikely to be effective.



(c)	Train	Test
Avg. # Objects (Things)	13.15	12.43
Avg. # Salient Objects	3.05	3.03
Avg. # Unique Object Categories	4.65	4.59
Avg. # Unique Salient Object Categories	1.78	1.80
Avg. # Regions (Stuff)	5.11	5.09

Figure 7.7: Further statistics of the proposed dataset. (a) and (b) reports the distribution of size and distance from image centre of salient objects, respectively. (c) average number of all objects (Things), salient objects and regions (Stuff). (d) shows an intensity map from overlays of all individual objects (i), salient objects (ii) and Stuff regions (iii). Local (e) and global (f) colour contrasts of salient objects.

7.3 Proposed Method

In this section, we first introduce the backbone (Section 7.3.1) of our network and discuss how contextual features are extracted and utilized. Then we specify how the proposed modules (Section 7.3.2 and Section 7.3.3) take advantage of the contextual features, in order to refine and augment features for saliency. Finally, we detail the Salient Instance Network (Section 7.3.4) for the task of salient object detection. An overview of the proposed framework is illustrated in Figure 7.8.

7.3.1 Backbone

Our network is built on the Mask-RCNN [15] architecture and we extract multi-scale features from the FPN [16]. We utilize the multi-scale features as input for 3 operations, namely, (1) object proposal, (2) context segmentation and (3) context feature refinement.

(1) Object Proposal: We apply RPN and RoIAlign [15] on the multi-scale features to generate object instance proposals and corresponding object features. This allows our network to perform saliency reasoning of individual objects.

(2) Context Segmentation: We include a Shared Context Segmentation Decoder for In-

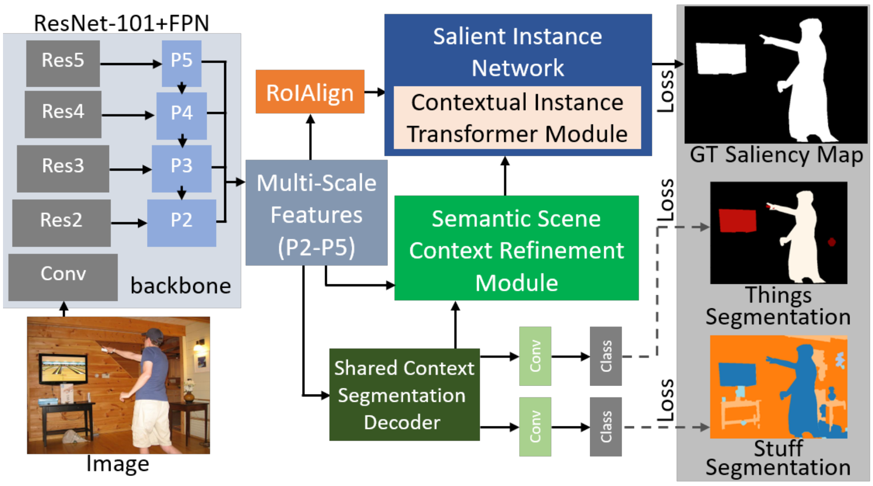


Figure 7.8: An overview of the proposed network. Our model extracts semantic features from the Shared Context Segmentation Decoder. The decoder is trained to reconstruct features for generating *Things* and *Stuff* categories. Our Semantic Scene Context Refinement (SSCR) module then utilizes the semantic features and multi-scale features to build the augmented scene context features, correlating the semantics of an image. Our Contextual Instance Transformer (CIT) module inside the Salient Instance Network, learns relationships between objects and scene context, and enhance saliency reasoning.

stance and Stuff Context Segmentation, in order to extract semantic context features for a given scene. The decoder takes the multi-scale features as input and reconstructs features for segmentation of *Things* and *Stuff* categories. From the decoder we extract semantic features $f^C \in \mathbb{R}^{\frac{W}{4}\times\frac{H}{4}\times 128}$, where $W \times H$ is the spatial dimensions for image I. The decoder follows similar design as in [133] to combine the multi-scale features for segmentation. The output features of the decoder are then passed through two separate convolutional layers for generating *Things* and then *Stuff* context segmentation.

(3) Context Feature Refinement: Third, we combine the multi-scale features with the features extracted from context segmentation **(2)**, producing the refined scene context features for boosting saliency reasoning.

These context features are used by the proposed SSCR (Section 7.3.2) and CIT (Section 7.3.3). SSCR builds the final scene context features by aggregating only useful context information. CIT learns the relationships between the scene context features and object features. The final salient object classification is detailed in Section 7.3.4.

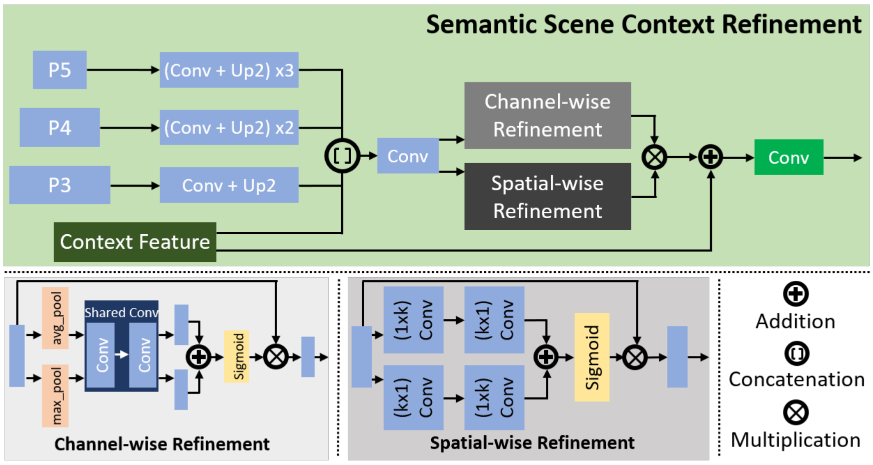


Figure 7.9: Details of the Semantic Scene Context Refinement (SSCR) module.

7.3.2 Semantic Scene Context Refinement (SSCR)

Previous works suggest that not all context information (e.g. , distractors) is relevant and useful to the final prediction task [82, 99, 134]. To address this problem we design this module to enhance the semantic information that has strong correlation to the scene context. This allows the network to augment contextual information learned only from saliency annotations with strong semantic scene context.

We build our semantic scene context features by refining the context features f^C obtained from the context segmentation decoder and multi-scale features (Figure 7.9). We only use feature levels [P3, P4, P5] from the multi-scale features, as these levels contain higher-level contextual features [25]. The three levels of multi-scale features are applied with operations similar to those in context segmentation, resulting in features $p^3, p^4, p^5 \in \mathbb{R}^{\frac{W}{4}\times\frac{H}{4}\times 128}$. We fuse these features into $f^F \in \mathbb{R}^{\frac{W}{4}\times\frac{H}{4}\times 128}$ by concatenating the multi-scale features with context features and applying a (1×1) convolutional layer. The concatenation helps suppress saliency distractors by utilising the scene context information. Next, we refine f^F in a channel-wise and spatial-wise manner.

Channel-wise Refinement: In CNN design, typically different semantic information are activated by distinct channel features [134, 135]. We include a channel-wise attention mechanism to weight channel features that are more relevant to semantic information. Given a set of features, we apply average pooling, max pooling and two convolutional layers with a ReLU and sigmoid activation. We then multiply the processed features (i.e., C_a, C_m) with the original feature x as:

$$\begin{aligned} CR(x) &= x \times Sigmoid(C_a(x) + C_m(x)), \\ C_a(x) &= conv_2(ReLU(conv_1(avgpool(x), W_1)), W_2), \\ C_m(x) &= conv_2(ReLU(conv_1(maxpool(x), W_1)), W_2), \end{aligned} \tag{7.2}$$

where $x = f^F$. W_1 and W_2 represent the parameters of the two convolutional layers.

Spatial-wise Refinement: Similarly, we include spatial-wise attention that leverages useful spatial information. Given a set of features, we employ two sets of double convolutional layers with alternating kernels, where one set contains kernels {1×k, k×1} and the other contains {k×1, 1×k}. The resulting features from the two sets of convolutional layers are added and sigmoid activation is applied to generate a spatial attention map. We weight the original feature x with the attention maps (i.e., S_1, S_2) through multiplication:

$$\begin{aligned} SR(x) &= x \times Sigmoid(S_1(x) + S_2(x)), \\ S_1(x) &= x \times ReLU(conv_2(ReLU(conv_1(s, W_1)), W_2)), \\ S_2(x) &= x \times ReLU(conv_4(ReLU(conv_3(s, W_3)), W_4)), \end{aligned} \tag{7.3}$$

where (W_1, W_2) and (W_3, W_4) are the parameters of the two sets of convolutional layers, with respect to {1×k, k×1} and {k×1, 1×k} kernels. After performing channel- and spatial-wise refinement on the fused features f^F, we combine the two outputs with Hadamard Multiplication. The product is further fused with the original context features (f^C) by addition and a final convolution is applied. This generates our final semantic scene context features f_{sc}.

$$f_{sc} = ReLU(conv((CR(f^F) \times SR(f^F)) + f^C, W_{sc})), \tag{7.4}$$

where W_{sc} are the parameters of the final convolutional layer. The process enables the enhancement of context from saliency features and scene context features, which are learned from salient object detection and context segmentation, respectively.

7.3.3 Contextual Instance Transformer (CIT)

It is observed in the literature that scene context influences eye movements [131]. However, most existing salient object detection methods do not model such high level understanding and relationships, not to mention, guiding saliency prediction in complex real-world scenes. As previously shown in Figure 7.1, saliency of individual objects requires semantic information about other objects and scene context to infer the high-level relationships and to differentiate objects from distractors. This module aims to learn relationships between objects and scene context for saliency reasoning.

We adapt transformers [95] to learn the dependencies between individual object features and scene context features in object-to-object and object-to-context relationships. We divide the module into two parts (see Figure 7.10). The first part is designed to learn relationships among the objects only, whereas, the second part learns relationships between individual objects and scene context. We use a scaled dot product attention layer with a single head on both types of relationship:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d}})V, \tag{7.5}$$

where $\sqrt{d}$ refers to normalization based on the feature size. Q, K and V are matrices corresponding to Queries, Keys and Values. Specifically, Q is projected from object features, while K and V are generated from either object features or scene context features. Multiplication of Q and K, followed by a softmax, produces an output that represents the degree of correlation between the feature vectors in Q and K. This is then used to weight the information of objects represented by the latent features V:

$$\begin{aligned} T_{OO} &= Attention(W_{q1}F^{o'}, W_{k1}F^{o'}, W_{v1}F^{o'}), \\ T_{OC} &= Attention(W_{q2}F^{o'}, W_{k2}f_{sc}, W_{v2}f_{sc}), \end{aligned} \tag{7.6}$$

where T_{OO} and T_{OC} are attention features modeling object-to-object and object-to-context relationships. $W_{\{q1,q2\}}$, $W_{\{k1,k2\}}$ and $W_{\{v1,v2\}}$ are parameters of fully connected and convolutional layers for linear projection. $F^{o'}$ refers to object features from RoIAlign and one fully connected layer (Section 7.3.4). During the attention in T_{OC}, K and V are flattened to become 1-D vectors (same as the object features). We then apply a fully connected layer and residual connection to both attention features (T_{OO} and T_{OC}). For the residual connection applied to T_{OC}, we first average pool the scene context features (f_{sc}) to transform the features into a 1-D vector. Finally, the two object-to-object and object-to-context relationship features are concatenated for our subsequent saliency classification (Section 7.3.4).

7.3.4 Salient Instance Network

Salient Instance Network performs the main salient object classification task from input object features, allowing our method to perform saliency reasoning on the object-level. It adapts from the second stage of Mask-RCNN, which consists of networks for predicting object class,

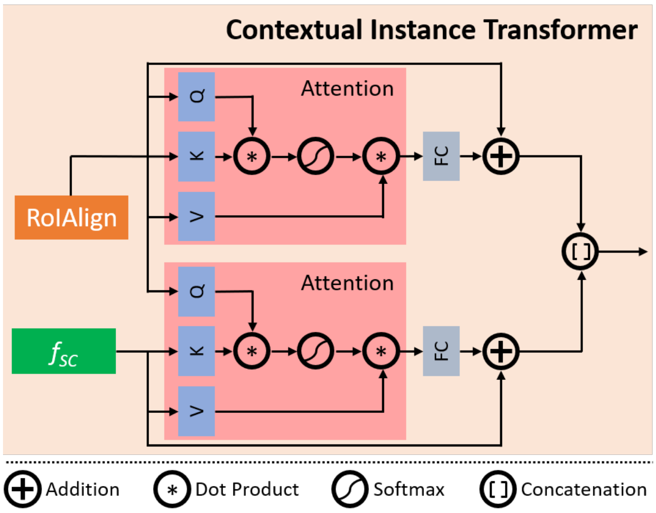


Figure 7.10: Details of the Contextual Instance Transformer (CIT) module.

bounding box and mask segmentation. We modify the network for salient object detection and enhance saliency prediction with scene context, visualized in Figure 7.11.

Our backbone (Section 7.3.1) generates candidate object features and predict their saliency. RPN and RoIAlign generate 2D features of individual object candidates. It is followed by a flatten and a fully connected layer. A feature vector, $f_i^o \in \mathbb{R}^{1024}$, is produced for each object, leading to a set of object features $F^o = \{f_1^o, f_2^o, \ldots, f_N^o\}$, where N=512 is the maximum number of object proposals. We obtain our final object features by fusing with object-to-object and object-to-context relationship information after employing CIT and two fully connected layers. A modified classification layer then determines the saliency of each object.

Most parts of our network share similar architecture and parameters with Mask-RCNN [15] and use the same loss functions for saliency prediction. We use the multi-class cross-entropy loss for training both instance and stuff context segmentation networks. The multi-class cross-entropy loss is defined as:

$$CE = -\sum_{c}^{C} \sum_{i}^{W \times H} y_i^k \log(p_i^k) + (1 - y_i^k) \log(1 - p_i^k), \tag{7.7}$$

where c is the class label in the set of classes C. i represents a pixel in an image with dimension $(W \times H)$. y_i^k and p_i^k are the ground-truth and predicted semantic segmentation maps indicating whether pixel i belongs to class c. Joint training with the semantic segmentation networks enables the saliency network to learn more useful semantic features relating to salient

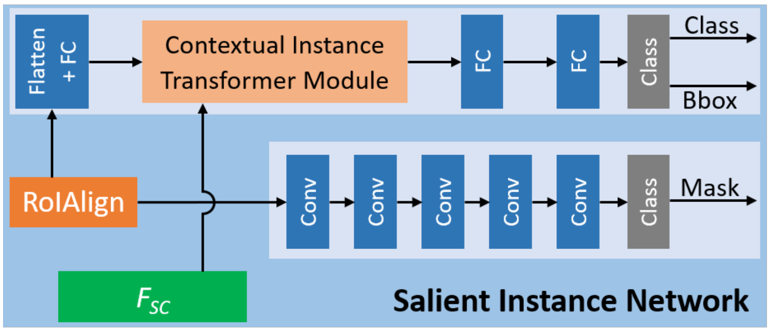


Figure 7.11: Details of the Salient Instance Network.

objects, compared to training with binary saliency labels only.

7.4 Experiments

For our experiments, we use ResNet-101 pre-trained on MS-COCO [27] as part of our backbone. Our model is based on the detectron2 framework [136] and is trained on a single NVIDIA GTX 1080 Ti GPU, for 30 epochs. A SGD optimizer with initial learning rate 0.001 is used, along with weight decay (10^{-4}) and momentum (0.9). We apply random cropping, flipping and multi-scale image training for data augmentation.

We carry out evaluation on our proposed dataset using a training set of 5,534 images for training and 2,554 images for testing.

We use three metrics namely, F-measure, Mean Absolute Error (MAE) and E-measure [137], to evaluate the performance of our model and state-of-the-arts. The F-measure provides a score of the overall performance in regards to the quality of the predicted saliency map. It is formulated by a weighted combination of Precision and Recall:

$$F_\beta = \frac{(1+\beta^2)Precision \times Recall}{\beta^2 Precision + Recall}, \tag{7.8}$$

where β^2=0.3. MAE calculates the average per-pixel disparity between predicted and ground-truth saliency maps. E-measure computes the pixel-level and image-level errors simultaneously to measure binary foreground similarities.

7.4.1 Comparison with State-of-the-Arts

We compare against 9 state-of-the-art methods in salient objects detection, including BASNet [18], CapSal [9], CPD-R [19], PFANet [25], EGNet [22], SCRN [20], ITSD [26], LDF [23]

Table 7.2: Quantitative comparison with state-of-the-art methods on our dataset. avgF refers to the average F-measure taken and E_m refers to E-measure. Red and blue indicate best and second best performances, respectively.

Method	avgF ↑	E_m ↑	MAE ↓
BASNet [18]	0.706	0.823	0.087
CapSal [9]	0.797	0.853	0.082
CPD-R [19]	0.803	0.854	0.071
PFANet [25]	0.676	0.772	0.131
S4Net [17]	0.625	0.720	0.149
EGNet [22]	0.815	0.863	0.067
SCRN [20]	0.786	0.842	0.076
ITSD [26]	0.776	0.854	0.070
LDF [24]	0.808	0.852	0.070
MINet [23]	0.810	0.861	0.067
Ours	0.849	0.872	0.062

and MINet [24]. Furthermore, we compare with S4Net [17] (salient instance segmentation), which also builds on the Mask-RCNN architecture like CapSal and our model. Note that in the comparison, CapSal is the only method not trained on our dataset (direct testing only with their pre-trained weights). CapSal requires GT captions data corresponding to GT saliency annotations. We also run into issues running their provided source code[2].

Quantitative Evaluation: We report the experimental results comparing the proposed model with state-of-the-arts in Table 7.2. It shows that our model quite significantly outperforms existing state-of-the-arts across all metrics. In particular, our model show substantial improvement in the average F-measure, with a performance increase of 4.17% over the second best method.

Qualitative Evaluation: We further showcase the performance of our model in Figure 7.12, which displays visual comparisons between our model and 10 state-of-the-art methods. Our model is able to correctly pick out unique and interesting salient objects among multiple distractors by utilizing the context of image scenes. This is often not the case for the other methods as they are unable to effectively distinguish between salient objects and distractors. The bottom row images further illustrate our model exploiting semantic information in order to fully segment the salient person from the bench. The other methods do not capture such semantic information. They suffer from additional false saliency on part of the bench or

[2] We tried the CapSal source code for pre-processing captions data (`https://github.com/zhangludl/code-and-dataset-for-CapSal`), but were unable to adapt their code for our dataset.

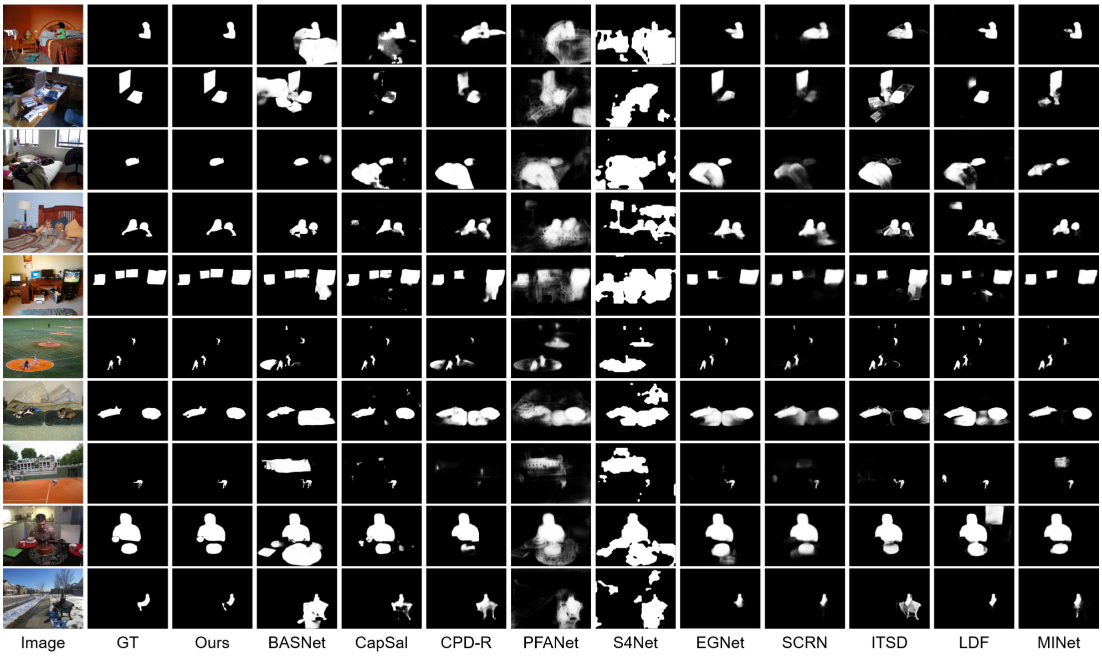


Figure 7.12: Qualitative comparison of the proposed method with ten state-of-the-art saliency methods: BASNet [18], CapSal [9], CPD-R [19], PFANet [25], S4Net [17], EGNet [22], SCRN [20], ITSD [26], LDF [24] and MINet [23].

unable to segment salient object correctly.

7.4.2 Comparison on Existing Datasets

Existing salient object datasets are not our target datasets for saliency, as they generally do not contain complex images and are not suitable for training our model (e.g. , no object instance and semantic segmentation annotations). Nevertheless, we provide further experiments to show the generalisability of our model on these existing datasets.

We carry out evaluation on five common benchmark datasets: ECSSD [10], PASCAL-S [11], HKU-IS [12], DUT-OMRON [13] and DUTS [14]. For fair comparison we use the training set (5534 images) *from our proposed dataset* to train all comparison models and *directly test* on the five datasets. Furthermore, the test images for each of the five datasets are filtered into a new subset that mainly include images that contain object categories defined in our dataset. These are the object categories our model is able to generate saliency prediction. The resulting ECSSD, PASCAL-S, HKU-IS, DUT-OMRON and DUTS thus respectively contains 928, 807, 4177, 3228 and 4338 test images.

Table 7.3: Quantitative comparison with state-of-the-art methods on five existing datasets to show *generalisability*. Note that we *train all comparison methods on our proposed dataset*, and *test directly* on these existing datasets for fair comparison. Furthermore, the test images for each dataset are filtered into a new subset that mainly include images that contain object categories defined in the proposed dataset. The number of images of the new subsets and the original test sets are respectively indicated next to the dataset. avgF refers to the average F-measure taken and E_m refers to E-measure. Red, Blue and Magenta respectively indicate the top 3 performance.

	ECSSD (928/1000)			PASCAL-S (807/850)			HKU-IS (4177/4447)			DUT-OMRON (3228/5168)			DUTS (4338/5019)		
Method	avgF ↑	E_m ↑	MAE ↓	avgF ↑	E_m ↑	MAE ↓	avgF ↑	E_m ↑	MAE ↓	avgF ↑	E_m ↑	MAE ↓	avgF ↑	E_m ↑	MAE ↓
BASNet [18]	0.748	0.834	0.092	0.740	0.827	0.092	0.800	0.884	0.073	0.736	0.859	0.071	0.694	0.825	0.085
CPD-R [19]	0.872	0.894	0.058	0.837	0.859	0.069	0.862	0.925	0.049	0.764	0.876	0.060	0.781	0.872	0.058
PFANet [25]	0.769	0.848	0.111	0.730	0.803	0.120	0.746	0.859	0.098	0.595	0.756	0.111	0.609	0.758	0.112
S4Net [17]	0.789	0.853	0.085	0.738	0.790	0.111	0.780	0.882	0.069	0.613	0.766	0.109	0.647	0.782	0.101
EGNet [22]	0.819	0.855	0.070	0.826	0.851	0.066	0.756	0.839	0.072	0.632	0.748	0.058	0.713	0.814	0.060
SCRN [20]	0.856	0.888	0.068	0.819	0.846	0.072	0.849	0.918	0.058	0.741	0.866	0.066	0.759	0.861	0.063
ITSD [26]	0.857	0.899	0.051	0.815	0.864	0.061	0.872	0.933	0.044	0.769	0.880	0.058	0.786	0.880	0.053
LDF [23]	0.875	0.888	0.057	0.848	0.858	0.062	0.875	0.927	0.046	0.749	0.860	0.068	0.787	0.869	0.057
MINet [24]	0.882	0.901	0.051	0.847	0.868	0.059	0.875	0.930	0.045	0.768	0.877	0.059	0.806	0.889	0.049
Ours	0.859	0.880	0.070	0.860	0.866	0.067	0.858	0.899	0.059	0.788	0.873	0.063	0.804	0.866	0.066

Quantitative Evaluation: Table 7.3 evaluates our technique on five common datasets in comparison to existing state-of-the-arts. We would like to note that the results reported in the table are established from the five subsets defined earlier. Figure 7.13 shows examples of simple image scenes with salient object categories that are not defined in our dataset. We also note here that CapSal [9] is omitted in this experiment as CapSal is unable to train on our dataset.

The results show that our proposed model is able to outperform state-of-the-arts for certain metrics and datasets with good margin. Particularly, we perform the best on the PASCAL-S for average F-measure and a very close second for E-measure. Our method also outperforms on DUT-OMRON in terms of average F-measure. For the rest, our model is able to produce quite comparable results to the best method for each dataset-metric combination.

Overall, our model can generalise quite well on common datasets, even when it is directly tested on those datasets. We believe that given sufficient data (i.e. , object instance and image segmentation), our model could be trained on existing datasets and thus potentially perform better.

Qualitative Evaluation: We present qualitative comparisons in Figure 7.14. The figure shows example images where our method is able to perform well, despite it not always achieving the best quantitative results. It demonstrates that our method is able to utilize scene context information, separate salient objects from surrounding background and reduce false saliency from distractors. This can be clearly seen in the third image for ECSSD, first image for HKU-IS, second image for DUT-OMRON and third image for DUTS. In these example images, our

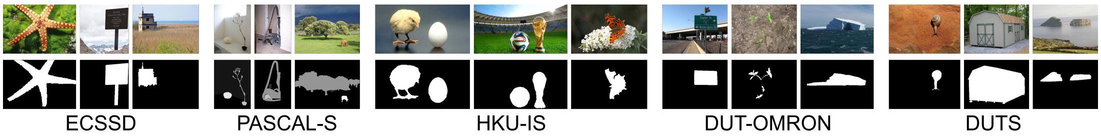


Figure 7.13: Example images in common datasets (ECSSD [10], PASCAL-S [11], HKU-IS [12], DUT-OMRON [13] and DUTS [14]) that contain simple scenes. There are only one or very few objects. The object categories are not defined in MS-COCO [27] and the datasets do not contain semantic segmentation annotations. As a result, our network is unable to train on these datasets and direct testing does not always lead to our network achieving new state-of-the-art performances (Table 7.3). Images (Top) and corresponding ground-truth saliency (Bottom).

Table 7.4: Ablation study of the proposed model on our dataset. Base: Mask-RCNN architecture, ISCS: Instance/Stuff Context Segmentation, SSCR: Semantic Scene Context Refinement, CIT: Contextual Instance Transformer.

Method	avgF ↑	E_m ↑	MAE ↓
Base	0.826	0.851	0.069
Base+ISCS	0.841	0.866	0.063
Base+ISCS+SSCR	0.845	0.869	**0.062**
Base+ISCS+CIT	**0.849**	0.871	**0.062**
Base+ISCS+SSCR+CIT	**0.849**	**0.872**	**0.062**

network is able to segment out the salient objects accurately, while existing techniques also predict false saliency of regions that surround the salient objects. Additionally, the figure reveals that our method has potential to further improve prediction if trained properly with sufficient data.

7.4.3 Ablation Study

We perform additional experiments to evaluate the effectiveness of our proposed modules. These results are shown in Table 7.4. It shows that the proposed modules produce improvements to the baseline saliency network. Our full model achieves the best overall performance and state-of-the-art results. This suggests that the proposed modules are able to effectively extract and enhance scene context information, then integrate them for saliency reasoning.

7.5 Limitations and Future Work

We find three reasons for our method not always outperforming on existing datasets: a) our method requires instance data for training. However, such data is not available in existing

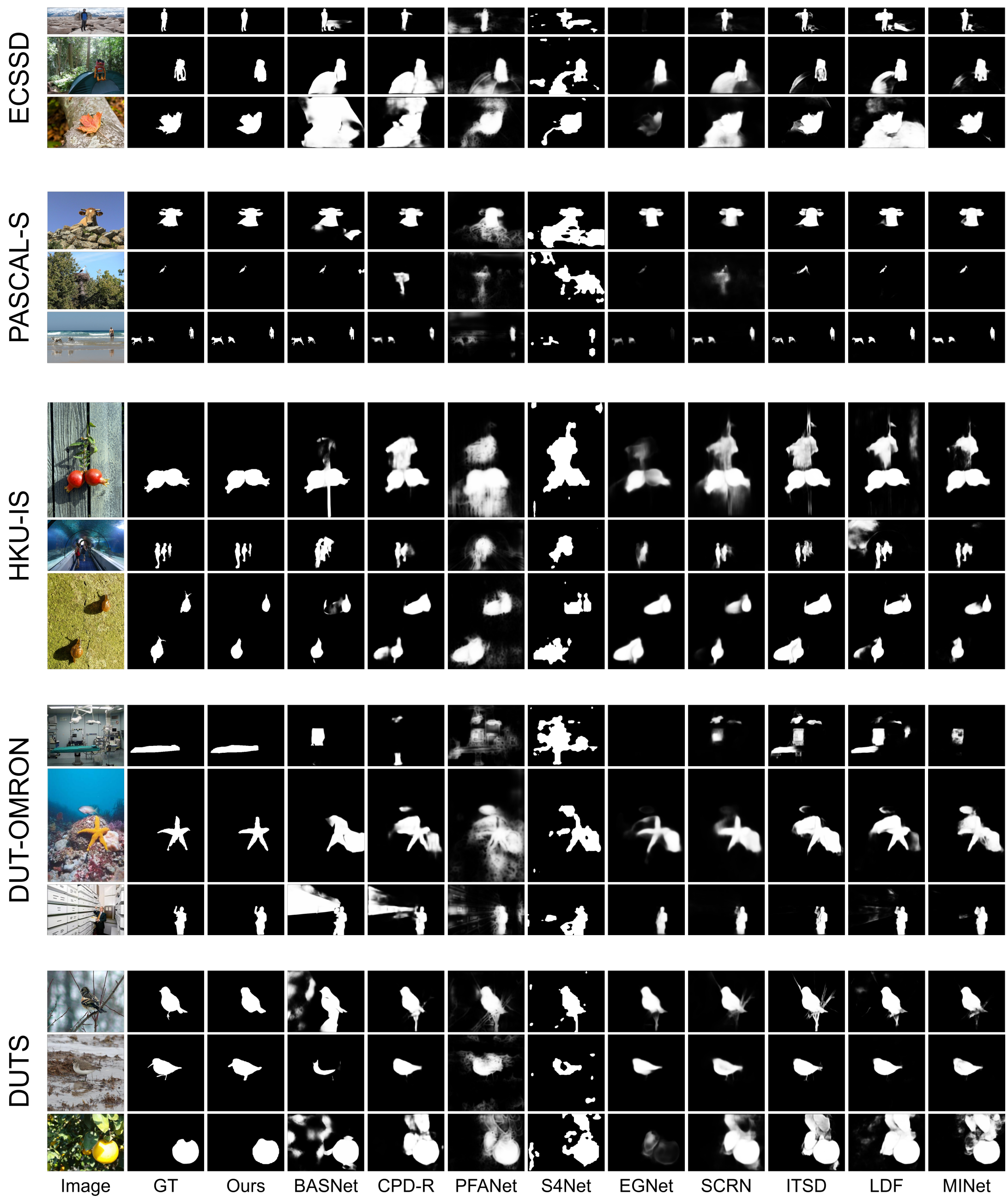


Figure 7.14: Visual comparison of our proposed method with state-of-the-arts (BASNet [18], CPD-R [19], PFANet [25], S4Net [17], EGNet [22], SCRN [20], ITSD [26], LDF [24] and MINet [23]) on existing salient object detection datasets (ECSSD [10], PASCAL-S [11], HKU-IS [12], DUT-OMRON [13] and DUTS [14]).

datasets. b) our method mainly focuses on multiple salient object detection in complex scenes. As the other datasets contain mostly images of very few objects, it is difficult for our model to explore/leverage object and scene context relationships. c) some test images also contain GT salient objects of categories not defined in our dataset. Our method thus may not recognise such objects. Despite the above constraints, our method still show some ability to capture and predict parts of those undefined objects, albeit not always outperforms. In the next Chapter 8, we aim to tackle these limitations by introducing a pixel-based saliency stream to mitigate failure of missing ground-truth salient objects.

7.6 Summary

In this chapter, we observed that existing salient object detection methods do not fully capture the semantic context of complex image scenes, leading them to produce false saliency of distractors and missing prediction of salient objects with relations to the scene context. We have also found that popular saliency benchmark datasets mostly contain images of simple scene structure, and do not provide real-world scenarios involving complex scenes with rich context. We have tackled these problems by proposing a new challenging dataset with complex scenes and a saliency model that exploits semantic scene context for improving saliency reasoning. Experimental results show that the proposed model outperforms state-of-the-art methods on the proposed dataset.

Chapter 8

Combining Instance-based and Pixel-based Saliency

Contents

8.1 Introduction **90**

8.2 Proposed Method **91**

8.2.1 Network Architecture Overview 91

8.2.2 Pixel-based Saliency Stream 92

8.2.3 Contextual Saliency Aggregation 93

8.3 Experiments **94**

8.3.1 Comparison with State-of-the-Arts 94

8.3.2 Ablation Study 96

8.4 Limitations and Future Work **97**

8.5 Summary **98**

8.1 Introduction

In this chapter, we aim to improve the limitations of our salient object detection model from Chapter 7. As mentioned in the previous chapter, we observe three main issues with our original method that becomes evident with existing datasets. The first issue is that our method requires instance annotation data for training, which is not available in existing datasets. Therefore, we are unable to directly train on those datatsets. Second, our method is designed for complex image scenes containing many objects with rich context. On the other hand, existing datasets are mostly built from simple image scenes with few objects. The third issue is that the existing datasets also contain salient objects, whose object categories are not defined in our proposed dataset. These issues are primarily caused by the design of our method being based on object proposals, which require specialized instance data for training.

Missing detection of certain objects is a common problem with instance-based (object proposal) saliency techniques. This is especially the case for newer datasets, where those objects have not been defined previously, requiring modifications to the network and updated training with additional instance annotations. Furthermore, instance-based techniques may learn some saliency bias to specific object categories if those objects are usually found to be more ground-truth salient throughout the dataset.

Traditional pixel-based saliency techniques are quite resilient to the above issues, as they generally learn abstract-like representation of a mixture of object categories. In addition, pixel-based techniques do not require further specialized data and can train directly from the ground-truth saliency map. However, these strengths also lead to pixel-based techniques being weaker in localizing individual object features.

Motivated by the above limitations, in this work we propose to integrate a pixel-based saliency stream with our original instance-based saliency network (from Chapter 7) to complement each other. We design our pixel-based saliency stream to train on both salient and non-salient objects. It encourages to learn discriminative object features and to separate salient and non-salient object features. We then develop a Contextual Saliency Aggregation (CSA) module, which learns to dynamically combine the saliency outputs from the instance-based saliency stream and a pixel-based saliency stream.

Contributions: In this chapter, we build upon the work and limitations from the previous Chapter 7. The main contributions of this work include:

- We propose to combine instance- and pixel- based saliency outputs with a new Contextual Saliency Aggregation (CSA) module to improve the final saliency prediction.

- We perform further experiments to evaluate our new method with comparisons to state-of-the-art techniques.

The work in this chapter is in preparation for submission to[1].

The following sections detail the integration of pixel-based saliency with instance-based saliency, along with a saliency aggregation module in Section 8.2. Section 8.3 show experimental results of our new method and the limitations are discused in Section 8.4. Finally, we conclude the chapter in Section 8.5.

8.2 Proposed Method

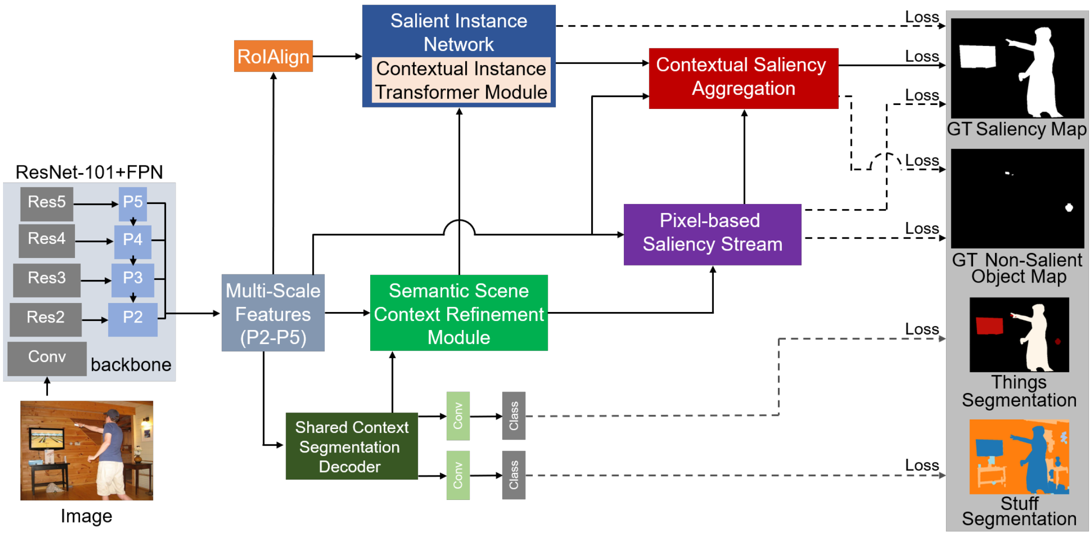

ResNet-101+FPN
Res5
Res4
Res3
Res2
Conv
P5
P4
P3
P2
backbone
Image
RoIAlign
Salient Instance Network
Contextual Instance Transformer Module
Contextual Saliency Aggregation
Multi-Scale Features (P2-P5)
Semantic Scene Context Refinement Module
Pixel-based Saliency Stream
Shared Context Segmentation Decoder
Conv
Class
Conv
Class
Loss
Loss
Loss
Loss
Loss
Loss
Loss
GT Saliency Map
GT Non-Salient Object Map
Things Segmentation
Stuff Segmentation

Figure 8.1: Architecture Overview. We extend our salient object detection method from Chapter 7. We introduce a pixel-based saliency stream and combine its prediction with our original instance-based saliency. The combination is performed by the Contexual Saliency Agggregation (CSA) module, which learns to dynamically fuse saliency predictions from the two streams.

8.2.1 Network Architecture Overview

We extend the salient object detection network from our preliminary work in Section 7.3. We incorporate a pixel-based saliency stream, which utilizes scene context features for generating pixel-level saliency prediction. It is introduced to complement the instance-based saliency

[1] Avishek Siris, Jianbo Jiao, Gary K.L. Tam, Xianghua Xie and Rynson W.H. Lau. "Scene Context-Aware Salient Object Detection". *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

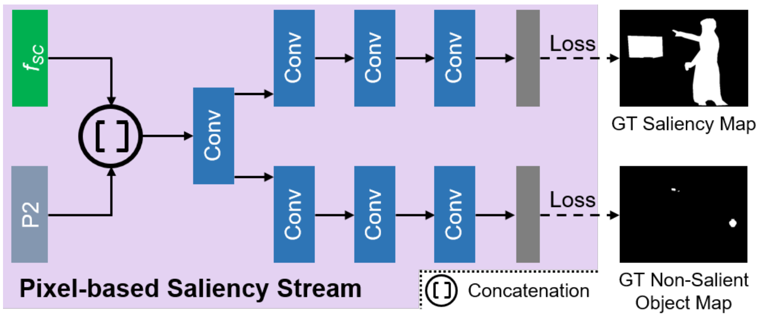


Figure 8.2: Details of the Pixel-based Saliency Stream.

prediction with finer segmentation accuracy, while reducing saliency bias towards specific object categories. Next, the outputs from the pixel-based saliency and instance-based saliency streams are fed as inputs to our Contexual Saliency Agggregation (CSA) module. CSA merges the predictions from the two saliency streams with scene context features, in order to produce the final saliency prediction. The aim of CSA is to generate the final saliency prediction by combining agreed saliency from the two saliency streams, at the same time recover saliency prediction missed by the other stream. It also enables suppression of false saliency by reasoning the saliency predictions between both saliency streams. An overview of the network archtecture is illustrated in Figure 8.1

8.2.2 Pixel-based Saliency Stream

To complement the predictions of our previous instance-based salient object detection method, we propose to integrate it with a pixel-based saliency stream. The goal of the pixel-based saliency stream is to promote detection of salient objects that is missed by the instance saliency, while also reducing saliency bias to common object categories. Figure 8.2 illustrates the structure of the pixel-based saliency stream.

We begin by combining the scene context feature (f_{sc}) with the low-level P2 feature from the multi-scale features, by using a concatenation and convolutional layer. This provides us with useful high-level contextual features of objects that is enhanced with lower-level details (e.g. , edge information) that is preserved by the P2 feature [22]. We then use this feature as input to two independent stacks of convolutional layers. Each stack consist of 3 convolutional layers, which is followed by a prediction layer. One of the stacks is designated for foreground salient object prediction, while the other stack is purposed for non-salient background object prediction. We design this approach for encouraging the network to explicitly learn features

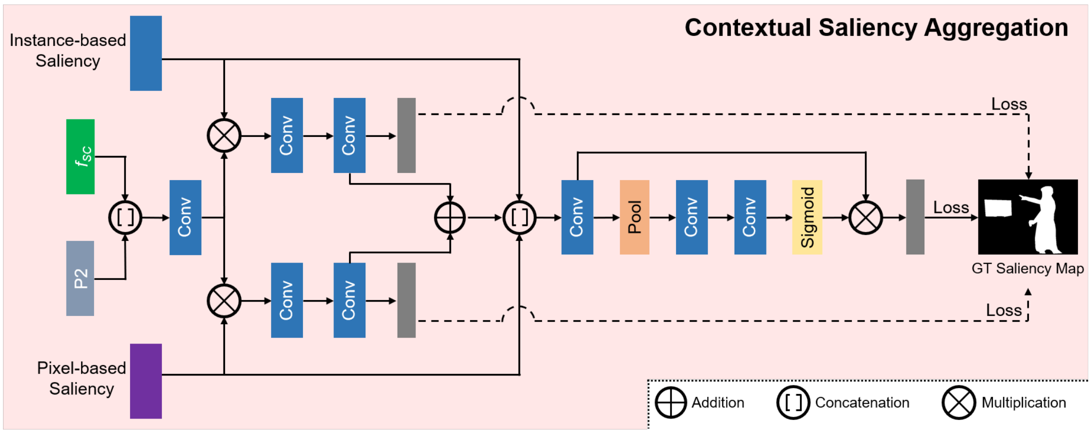


Figure 8.3: Details of the Contextual Saliency Aggregation (CSA) module.

that are able to distinguish from salient and non-salient object features.

8.2.3 Contextual Saliency Aggregation

Our original instance-based saliency is able to localize and predict saliency of indiviudal objects well. However, it may miss predictions or learn saliency bias towards specific object categories. The pixel-based saliency does not have much issues with missing detection of particular objects, nor will it learn strong bias towards some object categories. It is capable of learning salient features directly from the ground-truth saliency maps. But it lacks in proficiently segmenting individual salient objects without introducing false saliency from surrounding regions.

Here, we propose to combine the saliency outputs of the instance- and pixel- based saliency streams to produce the final saliency prediction (shown in Figure 8.3). Simply combining the saliency outputs from the two saliency streams may not be very effective, as there is no adaptive learning to reduce the propagation of errors from the two outputs. We instead take a progressive approach to the aggregation of the saliency outputs.

Similarly to the pixel-based saliency stream, we first apply a convolutional layer to the concatenation of the low-level P2 feature and scene context feature (f_{sc}). We then multiply this feature with the saliency outputs from the instance- and pixel- based saliency streams, generating two new saliency features that correspond to each of the saliency streams. It allows the network to obtain two saliency features conditioned on separate saliency processes, which can learn different context information and later combined for better saliency prediction. Next, we separately apply two convolutional layers resulting in the features f^{ins} and f^{pix}. We follow this with two prediction layers, which generate intermediate saliency predictions for the instance- and pixel- based saliency streams. This enables the progressive improvement of saliency predictions from our two saliency streams. We add together the features f^{ins} and f^{pix}, then concatenate it with the two intermediate saliency predictions and apply a convolutional layer. We then adopt channel-wise refinement (with similarities to Section 7.3.2) to allow the network to dynamically weight useful channel features that are derived from either the instance-based or pixel-based saliency features. Finally, we add a residual connection with the feature before the channel-wise refinement, after which a prediction layer is used to create the final saliency prediction.

We use binary cross-entropy loss to train the pixel-based saliency stream and the Contextual Saliency Aggregation module. The binary cross-entropy loss is defined similarly to Equation 7.7 (Section 7.3.4). However, it compares the ground-truth saliency map against the predicted saliency map. The ground-truth saliency map contains two class labels (Background = 0, Salient = 1), indicating non-salient or salient for certain pixels. The predicted saliency map contains the probability for pixels belonging to the salient class.

8.3 Experiments

We follow the same experimental settings performed in Section 7.4, but with minor changes. Our new model is trained on an RTX 3090 GPU to accommodate the increase in the network size. We train for 30 epochs with the SGD optimizer. The learning rate is set to 0.001 with weight decay (10^{-4}) and momentum (0.9). For data augmentation, we apply random cropping, flipping and multi-scale images.

Evaluation is carried out on our dataset, which was proposed in Section 7.2. We use a training set of 5,534 images and 2,554 images for testing.

Again we use the average F-measure (avgF), Mean Absolute Error (MAE) and E-measure for our evaluation metrics. Additionally, we introduce two more metrics, the max F-measure (maxF) and S-measure [138]. The S-measure simultaneously evaluates region-aware and object-aware structural similarity between the predicted and ground-truth saliency maps.

8.3.1 Comparison with State-of-the-Arts

We compare against 13 state-of-the-art methods in salient objects detection, including BASNet [18], CapSal [9], CPD-R [19], PFANet [25], S4Net [17], EGNet [22], SCRN [20], ITSD [26], LDF [23], MINet [24], CSNet [28], GateNet [29] and VST [30].

Table 8.1: Quantitative comparison with state-of-the-art methods on our dataset from Section 7.2. We also compare with our original model SCAS [3] from Chapter 7. avgF refers to the average F-measure, while maxF refers to the max F-measure. E_m refers to the E-measure and S_m to the S-measure. Red and blue indicate best and second best performances, respectively.

Method	avgF ↑	maxF ↑	E_m ↑	MAE ↓	S_m ↑
BASNet [18]	0.706	0.813	0.823	0.087	0.801
CapSal [9]	0.797	0.836	0.853	0.082	0.816
CPD-R [19]	0.803	0.844	0.854	0.071	0.829
PFANet [25]	0.676	0.782	0.772	0.131	0.764
S4Net [17]	0.625	0.644	0.720	0.149	0.678
EGNet [22]	0.815	0.862	0.863	0.067	0.846
SCRN [20]	0.786	0.860	0.842	0.076	0.841
ITSD [26]	0.776	0.860	0.854	0.070	0.844
LDF [24]	0.808	0.854	0.852	0.070	0.838
MINet [23]	0.810	0.859	0.861	0.067	0.838
CSNet [28]	0.692	0.802	0.802	0.111	0.789
GateNet [29]	0.811	0.867	0.860	0.065	0.852
VST [30]	0.794	0.844	0.840	0.072	0.834
SCAS [3]	0.849	0.861	0.872	0.062	0.828
Ours	0.838	0.875	0.875	0.062	0.857

As mentioned previously in Section 7.4.1, we are unable to train CapSal on our dataset, as we do not build caption data for our dataset and we run into issues with their code.

We provide further comparison against our original method SCAS [3] from the previous Chapter 7.

Quantitative Evaluation: Table 8.1 reports the comparison results between our new model and state-of-the-art techniques. The table shows that our new model has a better overall performance than our original network (SCAS) from Chapter 7. From the table, we can see that our new model achieves significant improvements on the max F-measure and the S-measure. Performance on the E-measure and MAE is quite comparable with minor increase and decrease, but we do lose some performance on the average F-measure.

Compared to the state-of-the-art techniques, our new model outperforms on all metrics. It produces performance gains of 0.92%, 2.82%, 1.39%, 4.61% and 0.85% to the second best methods, on the average F-measure, max F-measure, E-measure, MAE and S-measure, respectively.

Qualitative Evaluation: We showcase the visual comparison between our new model and state-of-the-arts in Figure 8.4. From the figure, we can see that our new model is able

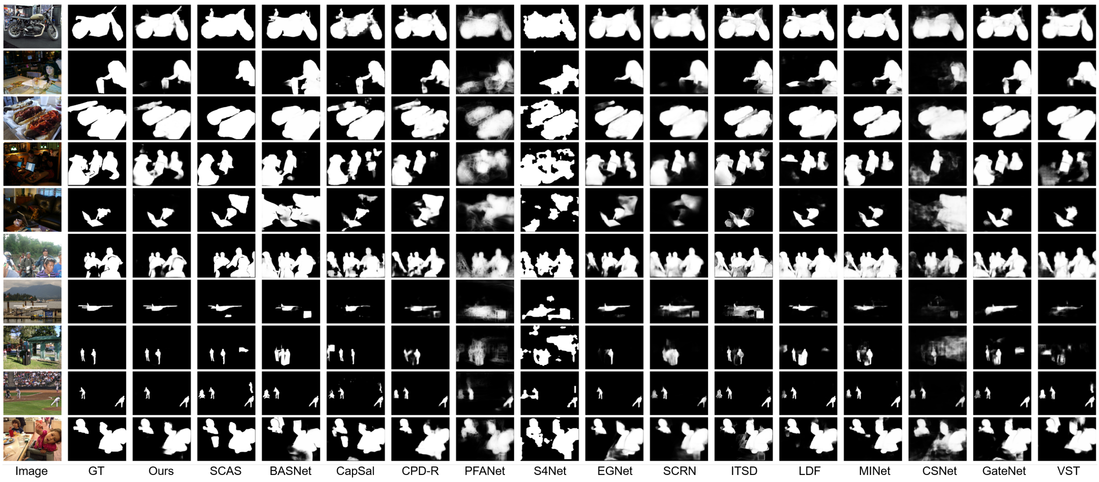


Figure 8.4: Qualitative comparison of the proposed method with state-of-the-art saliency methods: BASNet [18], CapSal [9], CPD-R [19], PFANet [25], S4Net [17], EGNet [22], SCRN [20], ITSD [26], LDF [24], MINet [23], CSNet [28], GateNet [29] and VST [30]. We provide further comparison with our original model, SCAS [3] (from Chapter 7).

to correctly identify multiple salient object in complex image scenarios. It is able to suppress multiple distractors by utilizing scene context information, while other state-of-the-art methods are unable to do so well. For instance, in the fifth row from Figure 8.4, many of the state-of-the-art methods do not confidently segment the salient dog and laptop. They also introduce false saliency of the couch, which has similar dark texture to the dog.

When comparing the new model with the original model (SCAS), the new model is able capture salient objects that SCAS misses (e.g. , second and third row). The integration of the pixel-based saliency stream enables the new model to capture better spatial low-level details, which improves the quality of segmentation in the final prediction. It also helps the network to focus on correct salient objects and remove false saliency of background objects.

8.3.2 Ablation Study

We further carry out an ablation study to show the effectiveness of the new modules, shown in Table 8.2. The table shows performance gains when adding the new modules, especially on the max F-measure and S-measure. The full model achieves the best overall results. This suggests that the integration of the pixel-based saliency stream with the contextual saliency aggregation, contributes to enhancing the prediction of the instance-based saliency.

Table 8.2: Ablation study of the proposed model on our dataset. Base: Mask-RCNN architecture, ISCS: Instance/Stuff Context Segmentation, SSCR: Semantic Scene Context Refinement, CIT: Contextual Instance Transformer, PsCSA: Pixel-based Saliency Stream with Contextual Saliency Aggregation

Method	avgF ↑	maxF ↑	E_m ↑	MAE ↓	S_m ↑
Base	0.826	0.839	0.851	0.069	0.819
Base+ISCS	0.841	0.853	0.866	0.063	0.825
Base+ISCS+SSCR	0.845	0.856	0.869	**0.062**	0.828
Base+ISCS+CIT	**0.849**	0.860	0.871	**0.062**	0.828
Base+ISCS+PsCSA	0.832	**0.875**	0.871	0.063	0.855
Base+ISCS+SSCR+CIT	**0.849**	0.861	0.872	**0.062**	0.828
Base+ISCS+SSCR+PsCSA	0.831	0.870	0.869	**0.062**	0.854
Base+ISCS+CIT+PsCSA	0.833	0.873	0.873	**0.062**	0.856
Base+ISCS+SSCR+CIT+PsCSA	0.838	**0.875**	**0.875**	**0.062**	**0.857**

8.4 Limitations and Future Work

We integrate saliency prediction from a pixel-based stream with the instance-based saliency stream, in order to support our network to capture missing salient objects. Although, our pixel-based saliency stream does help to predict missing salient objects, it can also introduce false saliency to small regions or parts of background objects. This is a typical issue with pixel-based saliency, as it performs saliency reasoning on the pixel/patch-level, which makes the scope of learning salient features to be very localized towards parts of an object. Therefore, it results in false saliency being generated if background regions contain features similar to parts of salient objects.

Another potential limitation we find is that sometimes the segmentation of salient objects is not fully captured and uniform. There are some cases where the segmentation of objects contain very weak or missing saliency within the body of an object. This is likely caused by the pixel-based saliency stream deeming those parts of an object to be related to non-salient background features, thus decreasing their saliency value.

Further investigation of how to effectively combine pixel- and instance- based saliency would be a good research avenue for future work. [9] is the only other work that considers this approach, but it is has not been fully explored in the saliency literature. A possible solution to the above issues would be to incorporate more explicit interactions between the instance-level and pixel-level features for saliency reasoning, rather than just learning to weight the saliency outputs from the two streams.

8.5 Summary

In this chapter we built upon the limitations of our original salient object detection model from Chapter 7. We introduced a pixel-based saliency stream to complement the weaknesses of instance-based saliency, which involves missing prediction of uncommon object categories and learning bias towards particular objects. This is performed by a new Contextual Saliency Aggregation module, which learns to combine the saliency outputs from the pixel- and instance-based saliency streams. Our experimental results show that our new model achieves new state-of-the-art performance, while promoting prediction of missing salient objects, reducing false saliency of whole background objects and improving the overall quality of segmentation.

Chapter 9

Conclusions and Future Work

Contents

9.1 Conclusions **100**

9.2 Future Work **101**

9.1 Conclusions

In this thesis we have investigated recent deep learning techniques in salient object detection and saliency ranking. We then explored alternative methods for saliency, which were motivated by findings in psychological studies. We first examined the domain of saliency ranking. We found that initial work proposed to perform relative ranking of salient objects, based on the differences in saliency agreement between multiple observers. However, this led to cases where multiple objects could be given tied saliency ranks. In contrast, psychological and neuropsychological studies have found that the human visual system shifts attention from one stimuli to another, due to the limited capacity of the human brain. Incorporating such attention shift for saliency ranking had not been explored previously. Motivated by these findings, we defined a new task of saliency ranking based on attention shift (in Chapter 5). We built a new dataset for salient object ranking based on attention shift, which was supported by our user study. We then proposed a deep learning model to predict saliency ranks with bottom-up and top-down attention mechanisms. Our experiments showed that deep learning techniques are able to model saliency ranking based on attention shift, by utilizing bottom-up and top-down approaches.

In Chapter 6, we extended our saliency rank network by improving upon its limitations. One of the limitations included our network being trained in two stages, which prevented the backbone and object features from being fine-tuned with saliency rank training. Another limitation was that our saliency rank network was defined as rank-id classification during training, which required further post-processing to determine the final discrete ranking. Even though our network was able to achieve good results with these limitations, our network was still not optimally trained. We addressed these limitations and proposed new optimal techniques for enhancing the performance of our original saliency rank network. We modified our network and trained it in an end-to-end manner, while adopting a learning-to-rank technique for training. These modifications provided a better training solution for saliency ranking, as it enabled fine-tuning of all network features and the ability to effectively learn saliency differences between multiple objects. Additionally, we investigated the impact of poor boundary distinction between objects leading to weaker performance. To address this issue, we introduced a new salient edge network that was coupled with the mask segmentation network. It allowed both networks to mutually improve the segmentation and edges of salient objects. We then introduced a new saliency rank evaluation metric and with further experiments, we demonstrated that our network was able to achieve new state-of-the-art results.

Following after our work in saliency ranking, we surveyed deep learning methods in traditional salient object detection. During our survey, we discovered that current salient object detection methods had difficulties in correctly predicting saliency on complex images. We observed that most of the common salient object detection datasets were not very challenging, and only contained images with few objects and simple background. This does not reflect real-world images which contain many objects and complex background. Moreover, we observed that the state-of-the-art techniques were usually trained on class-agnostic binary saliency labels, which do not enable learning of semantic contextual information that could be useful for saliency reasoning. In Chapter 7, with further motivation from psychological studies suggesting that semantic scene context influence attention, we proposed a new salient object detection framework. The framework utilized semantic segmentation of *Things* and *Stuff* categories for extracting our semantic scene context information, then learning of the relationships between salient objects and scene context. To accommodate the training of our new framework, we built a new salient object detection dataset that consisted of complex images with rich context. Results from our experiments showed that we can learn discriminative semantic context to boost saliency prediction in challenging complex scenes.

Finally in Chapter 8, we further built upon our salient object detection framework. We found that our original method had many limitations, due to the nature of its design. We observed that our instance-based (object proposal) saliency method may miss saliency prediction, especially if the object had not been previously defined. Additionally, it may learn saliency bias to common object categories. To this end, we proposed to combine a pixel-based saliency stream with the instance-based saliency stream to complement the outputs of each other. The combination of both streams facilitated in improved saliency reasoning, as it exploited saliency information learnt from two different saliency processes. It enhanced segmentation quality, while recovering missed saliency predictions and suppressing false saliency predictions. Our new salient object detection method was able to produce new state-of-the-art results and improve the weaknesses of our original framework.

9.2 Future Work

In this work we have observed many limitations and possible directions for future work. During our study in saliency ranking, we have identified that saliency ranking is still quite a new problem that has been under-explored, especially saliency ranking based on attention shift. We believe that saliency ranking can advance the problem of saliency detection. Not only does it model closely to the human visual attention process, but it would also provide great benefits to down-stream applications and problems that utilize saliency (e.g. , image compression, robotic navigation), as it offers rank priority information. One possible application could be to integrate task guided saliency detection and ranking for automated driving. In this case, the saliency method would learn to identify important salient regions or objects that relate to the task of driving. The identified regions or objects could be used as additional targets of interest to caution during automated driving, which may not be registered by hardware sensors or considered as a factor in simple driving scenarios and conditions. Ranking of the salient regions or objects could then specify order of importance that may be used to plan driving navigation and for reacting to certain events.

Furthermore, new saliency ranking techniques could be developed to incorporate better learning-to-rank approaches for improving the performance of current methods. Correctly predicting lower saliency ranks is a very challenging problem, since saliency differences between lower ranks become minuscule. A future work would be to explore for effective methods to detect minor saliency rank differences, as this is a limiting factor of our saliency rank network. It would be interesting to investigate how different scene compositions would affect the number of saliency ranks and the magnitude of attention that each rank would hold.

In terms of traditional salient object detection, the current saliency task is mainly designed on images with simple background and limited number of objects. This is not very useful as new research is heading towards complex tasks that require real-world complex images scenes. The saliency methods trained on simple images struggle to perform well on complex image scenarios and so, their use case in the pre-processing stage for complex problems would not be beneficial. We believe that the future research direction of saliency should focus more on complex image scenes. This would be more valuable and impactful to the vision research community, as it would instigate the development of more robust saliency techniques that would be more useful for complex down-stream applications.

To our knowledge, only a couple of works [9, 94] leverage high-level scene semantics for saliency detection. Future work would entail experimenting with various approaches to extract and employ semantic scene information. One possibility is to use classification of scene events to understand the context of the image scene, for weighting saliency of objects that highly relate to the event. It would also be interesting to utilize the understanding of object properties and their potential interactions with other objects for guiding saliency.

In regards to high-level semantics, graphs are very good at capturing structural relations. Deep learning has recently been applied to graphs, resulting in the emergence of Graph Convolutional Networks (GCNs) [139,140]. It facilitates the learning in the properties of graphical structures [141] and the interactions between multiple entities [142]. [2] exploit GCNs for relative saliency ranking, however, they mainly use person prior knowledge as the high-level semantic information for learning the interactions between salient objects. It would be interesting to learn more explicit semantic relationships between general objects and background regions, including the affordances of objects and the linguistic structure of a scene.

Additionally, further top-down attention factors have not been fully explored for saliency. Investigating how different tasks and goals of an observer can affect the saliency could be beneficial not only in understanding how humans behave, but also in developing general saliency methods that can adapt to various image/video settings (e.g. , different image scenes and tasks). It could assist in designing networks that could combine reasoning from different tasks, in order to perform a complex task. Other factors that could be studied may involve in how strongly observer preferences and interests may influence their attention, and whether intrinsic expectations of certain events direct attention.

Bibliography

[1] A. Siris, J. Jiao, G. K. Tam, X. Xie, and R. W. Lau, “Inferring attention shift ranks of objects for image saliency,” in *CVPR*, 2020, pp. 12 133–12 143.

[2] N. Liu, L. Li, W. Zhao, J. Han, and L. Shao, “Instance-level relative saliency ranking with graph reasoning,” *IEEE TPAMI*, 2021.

[3] A. Siris, J. Jiao, G. K. Tam, X. Xie, and R. W. Lau, “Scene context-aware salient object detection,” in *ICCV*, 2021, pp. 4156–4166.

[4] J. M. Wolfe, “Guided search 2.0 a revised model of visual search,” *Psychonomic bulletin & review*, vol. 1, no. 2, pp. 202–238, 1994.

[5] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[6] W. Wang, J. Shen, M.-M. Cheng, and L. Shao, “An iterative and cooperative top-down and bottom-up inference network for salient object detection,” in *CVPR*, 2019, pp. 5968–5977.

[7] A. Borji, M.-M. Cheng, Q. Hou, H. Jiang, and J. Li, “Salient object detection: A survey,” *Computational visual media*, vol. 5, no. 2, pp. 117–150, 2019.

[8] M. Amirul Islam, M. Kalash, and N. D. B. Bruce, “Revisiting salient object detection: Simultaneous detection, ranking, and subitizing of multiple salient objects,” in *CVPR*, 2018, pp. 7142–7150.

[9] L. Zhang, J. Zhang, Z. Lin, H. Lu, and Y. He, “Capsal: Leveraging captioning to boost semantics for salient object detection,” in *CVPR*, 2019, pp. 6024–6033.

[10] Q. Yan, L. Xu, J. Shi, and J. Jia, "Hierarchical saliency detection," in *CVPR*, 2013, pp. 1155–1162.

[11] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille, "The secrets of salient object segmentation," in *CVPR*, 2014, pp. 280–287.

[12] G. Li and Y. Yu, "Visual saliency based on multiscale deep features," in *CVPR*, 2015, pp. 5455–5463.

[13] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, "Saliency detection via graph-based manifold ranking," in *CVPR*, 2013, pp. 3166–3173.

[14] L. Wang, H. Lu, Y. Wang, M. Feng, D. Wang, B. Yin, and X. Ruan, "Learning to detect salient objects with image-level supervision," in *CVPR*, 2017, pp. 136–145.

[15] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *ICCV*, 2017, pp. 2980–2988.

[16] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *CVPR*, 2017, pp. 2117–2125.

[17] R. Fan, M.-M. Cheng, Q. Hou, T.-J. Mu, J. Wang, and S.-M. Hu, "S4net: Single stage salient-instance segmentation," in *CVPR*, 2019, pp. 6103–6112.

[18] X. Qin, Z. Zhang, C. Huang, C. Gao, M. Dehghan, and M. Jagersand, "Basnet: Boundary-aware salient object detection," in *CVPR*, 2019, pp. 7479–7489.

[19] Z. Wu, L. Su, and Q. Huang, "Cascaded partial decoder for fast and accurate salient object detection," in *CVPR*, 2019, pp. 3907–3916.

[20] ——, "Stacked cross refinement network for edge-aware salient object detection," in *ICCV*, 2019, pp. 7264–7273.

[21] H. Fang, D. Zhang, Y. Zhang, M. Chen, J. Li, Y. Hu, D. Cai, and X. He, "e," in *ICCV*, 2021, pp. 16 331–16 341.

[22] J.-X. Zhao, J.-J. Liu, D.-P. Fan, Y. Cao, J. Yang, and M.-M. Cheng, "Egnet: Edge guidance network for salient object detection," in *ICCV*, 2019, pp. 8779–8788.

[23] J. Wei, S. Wang, Z. Wu, C. Su, Q. Huang, and Q. Tian, "Label decoupling framework for salient object detection," in *CVPR*, 2020, pp. 13 025–13 034.

[24] Y. Pang, X. Zhao, L. Zhang, and H. Lu, "Multi-scale interactive network for salient object detection," in *CVPR*, 2020, pp. 9413–9422.

[25] T. Zhao and X. Wu, "Pyramid feature attention network for saliency detection," in *CVPR*, 2019, pp. 3085–3094.

[26] H. Zhou, X. Xie, J.-H. Lai, Z. Chen, and L. Yang, "Interactive two-stream decoder for accurate and fast saliency detection," in *CVPR*, 2020, pp. 9141–9150.

[27] T.-Y. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *ECCV*, 2014, pp. 740–755.

[28] S.-H. Gao, Y.-Q. Tan, M.-M. Cheng, C. Lu, Y. Chen, and S. Yan, "Highly efficient salient object detection with 100k parameters," in *ECCV*, 2020, pp. 702–721.

[29] X. Zhao, Y. Pang, L. Zhang, H. Lu, and L. Zhang, "Suppress and balance: A simple gated network for salient object detection," in *ECCV*, 2020, pp. 35–51.

[30] N. Liu, N. Zhang, K. Wan, L. Shao, and J. Han, "Visual saliency transformer," in *ICCV*, 2021, pp. 4722–4732.

[31] A. Borji and L. Itti, "State-of-the-art in visual attention modeling," *IEEE TPAMI*, vol. 35, no. 1, pp. 185–207, 2012.

[32] A. Borji, M.-M. Cheng, H. Jiang, and J. Li, "Salient object detection: A benchmark," *TIP*, vol. 24, no. 12, pp. 5706–5722, 2015.

[33] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *ICML*, 2015, pp. 2048–2057.

[34] S. Yu and D. Lisin, "Image compression based on visual saliency at individual scales," in *International Symposium on Visual Computing*, 2009, pp. 157–166.

[35] B. Lai and X. Gong, "Saliency guided dictionary learning for weakly-supervised image parsing," in *CVPR*, 2016, pp. 3630–3639.

[36] R. Zhao, W. Ouyang, and X. Wang, "Person re-identification by salience matching," in *ICCV*, 2013, pp. 2528–2535.

[37] ——, “Unsupervised salience learning for person re-identification,” in *CVPR*, 2013, pp. 3586–3593.

[38] W. Wang, J. Shen, R. Yang, and F. Porikli, “Saliency-aware video object segmentation,” *IEEE TPAMI*, vol. 40, no. 1, pp. 20–33, 2017.

[39] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *CVPR*, 2015, pp. 3431–3440.

[40] A. Borji, “Boosting bottom-up and top-down visual features for saliency estimation,” in *CVPR*, 2012, pp. 438–445.

[41] C. Koch and S. Ullman, “Shifts in selective visual attention: towards the underlying neural circuitry,” in *Matters of Intelligence*. Springer, 1987, pp. 115–141.

[42] L. Itti and C. Koch, “A saliency-based search mechanism for overt and covert shifts of visual attention,” *Vision Research*, vol. 40, no. 10-12, pp. 1489–1506, 2000.

[43] A. Palazzi, D. Abati, F. Solera, and R. Cucchiara, “Predicting the driver’s focus of attention: the dr (eye) ve project,” *IEEE TPAMI*, vol. 41, no. 7, pp. 1720–1733, 2018.

[44] G. Schillaci, S. Bodiroža, and V. V. Hafner, “Evaluating the effect of saliency detection and attention manipulation in human-robot interaction,” *International Journal of Social Robotics*, vol. 5, no. 1, pp. 139–152, 2013.

[45] U. Neisser, *Cognitive Psychology: Classic Edition*. Psychology Press, 2014.

[46] M.-S. Kim and K. R. Cave, “Top-down and bottom-up attentional control: On the nature of interference from a salient distractor,” *Perception & Psychophysics*, vol. 61, no. 6, pp. 1009–1023, 1999.

[47] R. Desimone and J. Duncan, “Neural mechanisms of selective visual attention,” *Annual Review of Neuroscience*, vol. 18, no. 1, pp. 193–222, 1995.

[48] F. Katsuki and C. Constantinidis, “Bottom-up and top-down attention: different processes and overlapping neural systems,” *The Neuroscientist*, vol. 20, no. 5, pp. 509–521, 2014.

[49] X. Hou and L. Zhang, “Saliency detection: A spectral residual approach,” in *CVPR*. Ieee, 2007, pp. 1–8.

[50] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum, “Learning to detect a salient object,” *IEEE TPAMI*, vol. 33, no. 2, pp. 353–367, 2010.

[51] N. Liu and J. Han, “Dhsnet: Deep hierarchical saliency network for salient object detection,” in *CVPR*, 2016, pp. 678–686.

[52] P. Zhang, D. Wang, H. Lu, H. Wang, and B. Yin, “Learning uncertain convolutional features for accurate saliency detection,” in *ICCV*, 2017, pp. 212–221.

[53] T. Judd, K. Ehinger, F. Durand, and A. Torralba, “Learning to predict where humans look,” in *ICCV*, 2009, pp. 2106–2113.

[54] B. W. Tatler, “The central fixation bias in scene viewing: Selecting an optimal viewing position independently of motor biases and image feature distributions,” *Journal of Vision*, vol. 7, no. 14, pp. 4–4, 2007.

[55] S. J. Russell, *Artificial intelligence a modern approach*. Pearson Education, Inc., 2010.

[56] S. Russell and P. Norvig, “Artificial intelligence: a modern approach,” 2002.

[57] Y. Bengio, I. Goodfellow, and A. Courville, *Deep learning*. MIT press Cambridge, MA, USA, 2017, vol. 1.

[58] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[59] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, “Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations,” in *ICML*, 2009, pp. 609–616.

[60] M. Xu, Y. Ren, and Z. Wang, “Learning to predict saliency on face images,” in *ICCV*, 2015, pp. 3907–3915.

[61] R. J. Peters and L. Itti, “Beyond bottom-up: Incorporating task-dependent influences into a computational model of spatial attention,” in *CVPR*, 2007, pp. 1–8.

[62] A. Borji, D. N. Sihite, and L. Itti, “Probabilistic learning of task-specific visual attention,” in *CVPR*, 2012, pp. 470–477.

[63] J. Pan, E. Sayrol, X. Giro-i Nieto, K. McGuinness, and N. E. O’Connor, “Shallow and deep convolutional networks for saliency prediction,” in *CVPR*, 2016, pp. 598–606.

[64] H. Song, W. Wang, S. Zhao, J. Shen, and K.-M. Lam, "Pyramid dilated deeper convlstm for video salient object detection," in *ECCV*, 2018, pp. 715–731.

[65] P. Zhang, D. Wang, H. Lu, H. Wang, and X. Ruan, "Amulet: Aggregating multi-level convolutional features for salient object detection," in *ICCV*, 2017, pp. 202–211.

[66] S. He, J. Jiao, X. Zhang, G. Han, and R. W. Lau, "Delving into salient object subitizing and detection," in *ICCV*, 2017, pp. 1059–1067.

[67] L. Zhang, J. Dai, H. Lu, Y. He, and G. Wang, "A bi-directional message passing model for salient object detection," in *CVPR*, 2018, pp. 1741–1750.

[68] L. Tang, B. Li, Y. Zhong, S. Ding, and M. Song, "Disentangled high quality salient object detection," in *ICCV*, 2021, pp. 3580–3590.

[69] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. H. Torr, "Deeply supervised salient object detection with short connections," in *CVPR*, 2017, pp. 3203–3212.

[70] W. Wang, J. Shen, X. Dong, and A. Borji, "Salient object detection driven by fixation prediction," in *CVPR*, 2018, pp. 1711–1720.

[71] Y. Zeng, Y. Zhuge, H. Lu, and L. Zhang, "Joint learning of saliency detection and weakly supervised semantic segmentation," in *ICCV*, 2019, pp. 7223–7233.

[72] A. Li, J. Zhang, Y. Lv, B. Liu, T. Zhang, and Y. Dai, "Uncertainty-aware joint salient object and camouflaged object detection," in *CVPR*, 2021, pp. 10 071–10 081.

[73] B. Aydemir, D. Bhattacharjee, S. Kim, T. Zhang, M. Salzmann, and S. Süsstrunk, "Modeling object dissimilarity for deep saliency prediction," *arXiv:2104.03864*, 2021.

[74] S. He, R. W. Lau, W. Liu, Z. Huang, and Q. Yang, "Supercnn: A superpixelwise convolutional neural network for salient object detection," *IJCV*, vol. 115, no. 3, pp. 330–344, 2015.

[75] R. Zhao, W. Ouyang, H. Li, and X. Wang, "Saliency detection by multi-context deep learning," in *CVPR*, 2015, pp. 1265–1274.

[76] G. Lee, Y.-W. Tai, and J. Kim, "Deep saliency with encoded low level distance map and high level features," in *CVPR*, 2016, pp. 660–668.

[77] Y. Wei, F. Wen, W. Zhu, and J. Sun, “Geodesic saliency using background priors,” in *ECCV*, 2012, pp. 29–42.

[78] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S.-M. Hu, “Global contrast based salient region detection,” *IEEE TPAMI*, vol. 37, no. 3, pp. 569–582, 2014.

[79] J. Zhang, X. Yu, A. Li, P. Song, B. Liu, and Y. Dai, “Weakly-supervised salient object detection via scribble annotations,” in *CVPR*, 2020, pp. 12 546–12 555.

[80] J. Zhang, J. Xie, and N. Barnes, “Learning noise-aware encoder-decoder from noisy labels by alternating back-propagation for saliency detection,” in *ECCV*, 2020, pp. 349–366.

[81] J. Su, J. Li, Y. Zhang, C. Xia, and Y. Tian, “Selectivity or invariance: Boundary-aware salient object detection,” in *ICCV*, 2019, pp. 3799–3808.

[82] T. Wang, L. Zhang, S. Wang, H. Lu, G. Yang, X. Ruan, and A. Borji, “Detect globally, refine locally: A novel approach to saliency detection,” in *CVPR*, 2018, pp. 3127–3135.

[83] Y. Zeng, P. Zhang, J. Zhang, Z. Lin, and H. Lu, “Towards high-resolution salient object detection,” in *ICCV*, 2019, pp. 7234–7243.

[84] Z. Chen, Q. Xu, R. Cong, and Q. Huang, “Global context-aware progressive aggregation network for salient object detection,” in *AAAI*, vol. 34, no. 07, 2020, pp. 10 599–10 606.

[85] Z. Luo, A. Mishra, A. Achkar, J. Eichel, S. Li, and P.-M. Jodoin, “Non-local deep features for salient object detection,” in *CVPR*, 2017, pp. 6609–6617.

[86] Y. Liu, J. Han, Q. Zhang, and C. Shan, “Deep salient object detection with contextual information guidance,” *TIP*, vol. 29, pp. 360–374, 2019.

[87] T. Wang, A. Borji, L. Zhang, P. Zhang, and H. Lu, “A stagewise refinement model for detecting salient objects in images,” in *ICCV*, 2017, pp. 4019–4028.

[88] J.-J. Liu, Q. Hou, M.-M. Cheng, J. Feng, and J. Jiang, “A simple pooling-based design for real-time salient object detection,” in *CVPR*, 2019, pp. 3917–3926.

[89] M. Kalash, M. A. Islam, and N. D. Bruce, “Relative saliency and ranking: Models, metrics, data and benchmarks,” *IEEE TPAMI*, vol. 43, no. 1, pp. 204–219, 2019.

[90] L. Zhang, C. Yang, H. Lu, X. Ruan, and M.-H. Yang, “Ranking saliency,” *IEEE TPAMI*, vol. 39, pp. 1892–1904, 2016.

[91] J. Li, D. Xu, and W. Gao, “Removing label ambiguity in learning-based visual saliency estimation,” *TIP*, vol. 21, no. 4, pp. 1513–1525, 2012.

[92] A. Torralba, “Modeling global scene factors in attention,” *JOSA A*, vol. 20, no. 7, pp. 1407–1418, 2003.

[93] S. Goferman, L. Zelnik-Manor, and A. Tal, “Context-aware saliency detection,” *IEEE TPAMI*, vol. 34, no. 10, pp. 1915–1926, 2011.

[94] N. Liu and J. Han, “A deep spatial contextual long-term recurrent convolutional network for saliency detection,” *TIP*, vol. 27, no. 7, pp. 3264–3274, 2018.

[95] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *NIPS*, 2017, pp. 5998–6008.

[96] C.-Y. Ma, A. Kadav, I. Melvin, Z. Kira, G. AlRegib, and H. Peter Graf, “Attend and interact: Higher-order object interactions for video understanding,” in *CVPR*, 2018, pp. 6790–6800.

[97] J. Jiao, Y. Wei, Z. Jie, H. Shi, R. W. Lau, and T. S. Huang, “Geometry-aware distillation for indoor semantic segmentation,” in *CVPR*, 2019, pp. 2869–2878.

[98] J. Kuen, Z. Wang, and G. Wang, “Recurrent attentional networks for saliency detection,” in *CVPR*, 2016, pp. 3668–3677.

[99] N. Liu, J. Han, and M.-H. Yang, “Picanet: Learning pixel-wise contextual attention for saliency detection,” in *CVPR*, 2018, pp. 3089–3098.

[100] Y. Piao, W. Ji, J. Li, M. Zhang, and H. Lu, “Depth-induced multi-scale recurrent attention network for saliency detection,” in *ICCV*, 2019, pp. 7254–7263.

[101] X. Hu, C.-W. Fu, L. Zhu, T. Wang, and P.-A. Heng, “Sac-net: spatial attenuation context for salient object detection,” *IEEE TCSVT*, 2020.

[102] S. Alpert, M. Galun, A. Brandt, and R. Basri, “Image segmentation by probabilistic bottom-up aggregation and cue integration,” *IEEE TPAMI*, vol. 34, no. 2, pp. 315–327, 2011.

[103] V. Movahedi and J. H. Elder, "Design and perceptual validation of performance measures for salient object segmentation," in *CVPR Workshops*, 2010, pp. 49–56.

[104] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *ICCV*, 2001, pp. 416–423.

[105] M. Jiang, S. Huang, J. Duan, and Q. Zhao, "Salicon: Saliency in context," in *CVPR*, 2015, pp. 1072–1080.

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

[107] C.-K. Chang, C. Siagian, and L. Itti, "Mobile robot vision navigation & localization using gist and saliency," in *IROS*. IEEE, 2010, pp. 4147–4154.

[108] Q. Zheng, J. Jiao, Y. Cao, and R. W. Lau, "Task-driven webpage saliency," in *ECCV*, 2018, pp. 287–302.

[109] A. Recasens, A. Khosla, C. Vondrick, and A. Torralba, "Where are they looking?" in *NIPS*, 2015, pp. 199–207.

[110] S. Gorji and J. J. Clark, "Attentional push: A deep convolutional network for augmenting image salience with shared attention modeling in social scenes," in *CVPR*, 2017, pp. 2510–2519.

[111] L. Zhang, M. H. Tong, T. K. Marks, H. Shan, and G. W. Cottrell, "Sun: A bayesian framework for saliency using natural statistics," *Journal of Vision*, vol. 8, no. 7, pp. 32–32, 2008.

[112] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE TPAMI*, no. 11, pp. 1254–1259, 1998.

[113] A. Borji and L. Itti, "Exploiting local and global patch rarities for saliency detection," in *CVPR*, 2012, pp. 478–485.

[114] W. Einhäuser, M. Spain, and P. Perona, "Objects predict fixations better than early saliency," *Journal of Vision*, vol. 8, no. 14, pp. 18–18, 2008.

[115] D. Gao, S. Han, and N. Vasconcelos, "Discriminant saliency, the detection of suspicious coincidences, and applications to visual recognition," *IEEE TPAMI*, vol. 31, no. 6, pp. 989–1005, 2009.

[116] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *CVPR*, 2018, pp. 6077–6086.

[117] J. Xu, M. Jiang, S. Wang, M. S. Kankanhalli, and Q. Zhao, "Predicting human gaze beyond pixels," *Journal of Vision*, vol. 14, no. 1, pp. 28–28, 2014.

[118] J. Feng, Y. Wei, L. Tao, C. Zhang, and J. Sun, "Salient object detection by composition," in *ICCV*, 2011, pp. 1028–1035.

[119] P. Siva, C. Russell, T. Xiang, and L. Agapito, "Looking beyond the image: Unsupervised learning for object saliency and detection," in *CVPR*, 2013, pp. 3238–3245.

[120] J. Zhang, S. Sclaroff, Z. Lin, X. Shen, B. Price, and R. Mech, "Unconstrained salient object detection via proposal subset optimization," in *CVPR*, 2016, pp. 5733–5742.

[121] C. Cao, X. Liu, Y. Yang, Y. Yu, J. Wang, Z. Wang, Y. Huang, L. Wang, C. Huang, W. Xu *et al.*, "Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks," in *ICCV*, 2015, pp. 2956–2964.

[122] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola, "Stacked attention networks for image question answering," in *CVPR*, 2016, pp. 21–29.

[123] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual attention network for image classification," in *CVPR*, 2017, pp. 3156–3164.

[124] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *CVPR*, 2018, pp. 7794–7803.

[125] A. Torralba, "Modeling global scene factors in attention," *JOSA A*, vol. 20 7, pp. 1407–18, 2003.

[126] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[127] T. Cheng, X. Wang, L. Huang, and W. Liu, “Boundary-preserving mask r-cnn,” in *ECCV*, 2020, pp. 660–676.

[128] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li, “Learning to rank: from pairwise approach to listwise approach,” in *ICML*, 2007, pp. 129–136.

[129] F. Xia, T.-Y. Liu, J. Wang, W. Zhang, and H. Li, “Listwise approach to learning to rank: theory and algorithm,” in *ICML*, 2008, pp. 1192–1199.

[130] Y. Wang, L. Wang, Y. Li, D. He, and T.-Y. Liu, “A theoretical analysis of ndcg type ranking measures,” in *COLT*. PMLR, 2013, pp. 25–54.

[131] A. Torralba, A. Oliva, M. S. Castelhano, and J. M. Henderson, “Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search.” *Psychological review*, vol. 113, no. 4, p. 766, 2006.

[132] C. Fosco, A. Newman, P. Sukhum, Y. B. Zhang, N. Zhao, A. Oliva, and Z. Bylinskii, “How much time do you have? modeling multi-duration saliency,” in *CVPR*, 2020, pp. 4473–4482.

[133] A. Kirillov, R. Girshick, K. He, and P. Dollár, “Panoptic feature pyramid networks,” in *CVPR*, 2019, pp. 6399–6408.

[134] X. Zhang, T. Wang, J. Qi, H. Lu, and G. Wang, “Progressive attention guided recurrent network for salient object detection,” in *CVPR*, 2018, pp. 714–722.

[135] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, and T.-S. Chua, “Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning,” in *CVPR*, 2017, pp. 5659–5667.

[136] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, “Detectron2,” https://github.com/facebookresearch/detectron2, 2019.

[137] D.-P. Fan, C. Gong, Y. Cao, B. Ren, M.-M. Cheng, and A. Borji, “Enhanced-alignment measure for binary foreground map evaluation,” in *IJCAI*, 2018.

[138] D.-P. Fan, M.-M. Cheng, Y. Liu, T. Li, and A. Borji, “Structure-measure: A new way to evaluate foreground maps,” in *ICCV*, 2017, pp. 4548–4557.

[139] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering,” *NIPS*, vol. 29, 2016.

[140] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *ICLR*, 2017.

[141] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, “Graph attention networks,” *arXiv preprint arXiv:1710.10903*, 2017.

[142] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh, “Graph r-cnn for scene graph generation,” in *ECCV*, 2018, pp. 670–685.