University of Nottingham
UK | CHINA | MALAYSIA

Learning to Rank Salient Objects using Transformers and Graph Reasoning

Bowen Deng
20206413

Supervised by

Prof. Michael Pound

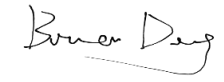
Prof. Andrew French

Thesis submitted to the University of Nottingham

for the degree of Doctor of Philosophy

I hereby declare that this dissertation is all my own work

except as indicated in the text:

Signature 

Date 24 / 09 / 2023

Abstract

This thesis explores the domain of salient object detection, aiming to find the most visually important objects within a given image. Many of the current approaches have focused on datasets with many images containing only a single salient object located towards the center. We focus here on the more complex task of images containing multiple objects, where relative saliency between objects must also be evaluated. A novel multiple salient object detection framework is proposed, utilizing both spatial and channel-wise non-local blocks within a convolutional network. The experiments compare the approach against 14 state-of-the-art methods on five widely used SOD benchmarks and a newly curated multi-object dataset. The proposed method exceeds all previous state-of-the-art approaches in three evaluation metrics and provides a further performance boost against competing techniques on the proposed dataset.

We then build upon this work to investigate the multiple salient object detection task in greater depth, exploring the problem of instance-level relative saliency ranking. This is an emerging field, and considering the lack of appropriate datasets in this domain, we produce a large-scale instance-level relative saliency ranking dataset using real human fixations. To the best of our knowledge, this is the first and largest dataset created by real human fixations for relative saliency ranking. A novel framework is then introduced that models multi-scale ranking-aware information cues in a nested style graph, drawing features from a query-based transformer. Experimental findings demonstrate the effectiveness of this proposed method. We exceed all previous state-of-the-art approaches with a large margin under three evaluation metrics. The model and full dataset will be released into the community.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Prof. Michael Pound, for his unwavering support and guidance throughout my Ph.D. journey. Prof. Michael Pound has been far more than a supervisor to me; he has been a mentor and a friend, making my last four years not just productive in research but also enriching on a personal level. His expertise in the field of computer vision has been invaluable to my research. He has consistently encouraged me to aim high, to venture into new areas, and to develop a strong sense of academic rigor. Beyond the academics, he has offered valuable support in other aspects of life that have made my journey smoother and more enjoyable. Therefore, I consider myself incredibly fortunate to have had Prof. Michael Pound as my supervisor.

I would also give special thanks to my second supervisor Prof. Andrew French for his continuous support and suggestions when undertaking my research.

Finally, I would like to thank University of Nottingham's Future Food Beacon, and the Biotechnology and Biological Sciences Research Council grant BB/T012129/1 for the funding source of my Ph.D.

Contents

Abstract i
Acknowledgements ii
Contents iii
List of Tables viii
List of Figures xi
Chapter 1 Introduction 1
1.1 Salient Object Detection 1
1.2 Relative Saliency Ranking 4
1.3 Objective and Contribution 7
1.4 Thesis Structure 8
1.5 Papers from This Thesis 9
Chapter 2 Background 10
2.1 Introduction 10
2.2 Semantic Segmentation 10
2.2.1 Fully Convolutional Neural Network 11
2.2.2 U-Net 11
2.3 Salient Object Detection 12
2.3.1 Deeply Supervised Salient Object Detection with Short Connections 13
2.3.2 A Bi-directional Message Passing Model for Salient Object Detection 15
2.3.3 EGNet: Edge Guidance Network for Salient Object Detection 17
2.3.4 PoolNet: A Simple Pooling-Based Design for Real-Time Salient Object Detection 19
2.3.5 Pyramid Feature Attention Network for Saliency Detection 20
2.3.6 Stacked Cross Refinement Network for Edge-Aware Salient

Object Detection 22
2.3.7 Summary of Popular SOD Models 25
2.4 Instance Segmentation 27
2.4.1 Mask R-CNN 28
2.4.2 CenterMask 30
2.4.3 SOLO 31
2.4.4 BlendMask 32
2.4.5 Summary of Popular Instance Segmentation Models 34
2.5 Relative Saliency Ranking 35
2.5.1 RSDNet 36
2.5.2 ASRNet 40
2.5.3 IRSR 43
2.5.4 SORNet 46
2.5.5 OCOR 48
2.5.6 Summary of RSR Models 49
2.6 Conclusion 50
Chapter 3 Multiple Salient Object Detection 52
3.1 Introduction 52
3.2 Background 53
3.2.1 Non-local Neural Networks 53
3.2.2 Dual Attention Network for Scene Segmentation 55
3.2.3 Discussion 57
3.3 Research Gaps - From Salient Object Detection to Multiple Salient Object Detection 58
3.4 Proposed Dataset 60
3.5 Proposed Method 63
3.5.1 Non-Local Guidance Module 64
3.5.2 Feature Fusion 67
3.5.3 Saliency Inference 70
3.6 Experiment 71

3.6.1 Datasets and Evaluation Metrics 71
3.6.2 Implementation Details 72
3.6.3 Quantitative Comparisons with the State-of-the-Art 73
3.6.4 Precision-Recall Curves Comparison 78
3.6.5 Visual Comparison 82
3.6.6 Ablation Studies 85
3.7 Conclusion 91
Chapter 4 Saliency Ranking Dataset 93
4.1 Introduction 93
4.2 Background 94
4.2.1 ASSR Dataset 95
4.2.2 IRSR Dataset 96
4.2.3 Summary of Saliency Ranking Dataset 98
4.3 Research Gaps 99
4.3.1 From Mouse-Trajectory based Fixations to Eye-Tracker based Fixations 99
4.3.2 From Fixed Number Salient Instances to Unlimited Instances 101
4.3.3 Image Selection and Instance Ranking Annotations 102
4.4 Data Collecting Strategy 103
4.4.1 Step 1: Image Selection 103
4.4.2 Step 2: Gaze Recording 104
4.4.3 Step 3: Fixation Filtering 105
4.4.4 Step 4: Salient Objects Threshold 108
4.4.5 Step 5: Annotating 111
4.4.6 Step 6: Ranking Assignments 114
4.4.7 Step 7: Ground Truth Maps 115
4.5 Dataset Structure and Examples 117
4.6 Statistics on Proposed Dataset 119
4.6.1 Instance Number Per Image 120

4.6.2 Instance Categories 121
4.6.3 Relative Saliency Ranking in Categories 123
4.6.4 Category Complexity 125
4.6.5 Foreground Size 127
4.6.6 Instance Size 130
4.6.7 Instance Location 131
4.6.8 Instance Contrast 133
4.7 Conclusion 135
Chapter 5 Exploration of Transformers for Salient Object Detection .. 136
5.1 Introduction 136
5.2 Background 137
5.2.1 An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale 137
5.2.2 DETR: End-to-End Object Detection with Transformers 140
5.2.3 Deformable DETR: Deformable Transformers for End-to-End Object Detection 142
5.2.4 CvT: Introducing Convolutions to Vision Transformers . 143
5.2.5 Swin Transformer: Hierarchical Vision Transformer using Shifted Windows 146
5.2.6 Mask2former: Masked-attention Mask Transformer for Universal Image Segmentation 148
5.2.7 Summary of Popular Transformer Models 150
5.3 Experiments 152
5.3.1 CvT-21 Backbone 152
5.3.2 CvT-21 backbone with Simple Decoder 153
5.3.3 CvT-21 backbone with reverse CvT-21 decoder 154
5.3.4 CvT-21 backbone with reverse CvT-21 decoder using channel-wise attention transformer 155
5.3.5 CvT-21 backbone with reverse CvT-21 decoder using channel-wise attention transformer and channel convolution for

upsampling 156
5.4 Conclusion 159
Chapter 6 Instance-Level Relative Saliency Ranking 161
6.1 Introduction 161
6.2 Background 161
6.2.1 Graph Neural Networks 161
6.2.2 Graph Attention Networks 164
6.2.3 Graph Convolutional Networks 166
6.2.4 Gated Graph ConvNet 167
6.2.5 Summary of Graph Reasoning Methods 168
6.3 Research Gaps 169
6.4 Query as Graph Network 171
6.4.1 Representative Aggregation Pathway 175
6.4.2 Global Representative Graph 176
6.4.3 Representative Feedback Pathway 177
6.4.4 Tri-tiered Nested Graph 178
6.4.5 Multi-layer QAGNet 179
6.5 Experiment 180
6.5.1 Datasets 180
6.5.2 Metrics 180
6.5.3 Implementation Details 183
6.5.4 Comparisons with the State-of-the-Art 184
6.5.5 Ablation Studies 191
6.6 Conclusion 195
Chapter 7 Conclusion 197
7.1 Contributions 197
7.2 Future Work 198
7.3 Potential Applications 200
Bibliography 202

List of Tables

Table 2-1 Summary of popular SOD models with main characteristics, advantages, potential disadvantages, and results. 26

Table 2-2 Summary of popular instance segmentation models with key features, advantages, potential disadvantages, and results. 35

Table 2-3 Summary of popular RSR models with key features, advantages, potential disadvantages, and results. 50

Table 3-1 Quantitative comparison with other state-of-the-art methods on 3 widely used relatively easy datasets. ↑ and ↓ indicate higher or lower is better respectively and * denotes weakly-supervised methods. The best three results among both backbones are marked as red, blue and cyan. Our method achieves top results under 3 evaluation metrics across all datasets without any pre-processing and post-processing. 74

Table 3-2 Quantitative comparison with other state-of-the-art methods on 2 widely used relatively hard datasets and the proposed MSOD dataset. ↑ and ↓ indicate higher or lower is better respectively and * denotes weakly-supervised methods. The best three results among both backbones are marked as red, blue and cyan. Our method achieves top results under 3 evaluation metrics across all datasets without any pre-processing and post-processing. 76

Table 3-3 Ablation analysis of different components in our proposed architecture. 85

Table 3-4 Performance comparison of different NLGM configurations. SSNLB and CSNLB refer to spatial-space non-local block and channel- space non-local block respectively. All three configurations are without FFG and ERM. 88

Table 3-5 Performance comparison of different NLGM architectures. All structures here are without FFG and ERM. 89

Table 4-1 Summary and comparison of current popular saliency ranking datasets ASSR and IRSR. 99
Table 4-2 Statistics on the foreground size. 128
Table 4-3 Statistics on the instance size. 131
Table 5-1 Summary and comparison of popular transformer techniques including key features, applications, contributions and potential limitations. 151
Table 5-2 Detailed architecture of CvT-21 proposed in [75]. Conv. Embed.: Convolutional Token Embedding. Conv. Proj.: Convolutional Projection. Hi and Di is the number of heads and embedding feature dimension in the ith MHSA module. Ri is the feature dimension expansion ratio in the ith MLP layer. 153
Table 5-3 Performance comparison between other state-of-the-art methods and the proposed architecture here on DUTOMRON dataset, the best result has been marked bold. 154
Table 5-4 Performance comparison between other state-of-the-art methods and the proposed architecture based on DUTOMRON dataset. Here, version 2 indicates the model that only encoder loads pretrained parameters, while version 3 denotes the one that both encoder and decoder load pretrained parameters. 155
Table 5-5 Performance comparison between other SOD methods and the proposed model (version 4) using channel-wise attention mechanism based on DUTOMRON dataset. 156
Table 5-6 The detailed configuration of the proposed architecture here. 158
Table 5-7 Two different channel-wise self-attention methods explored here. 159
Table 5-8 Performance comparison between other state-of-the-art methods and the proposed architecture with two different channel-wise self-attention methods based on DUTOMRON dataset. .. 159

Table 6-1 Quantitative Comparison with other saliency ranking methods. Different backbones are shown in the 2nd column, e.g., ResNet [28], VoVNet [92] and Swin [96]. We show our proposed method in ResNet-50, Swin-Base and Swin-large. The best two results have been marked as red and blue. For different methods, the number of parameters is shown in the last column. ↑ indicates the higher the better, while ↓ denotes the lower the better. Note _U and _L indicate the unlimited version and limited version models as illustrated in Section 6.5.4. 187
Table 6-2 Ablation analysis of different modules in our proposed method. 191
Table 6-3 Ablation analysis on the layer number and short connection of QAGNet. 193
Table 6-4 Ablation study on the number of salient instance queries. 194
Table 6-5 Ablation study on the initialization method for representative queries. 195

List of Figures

Figure 1-1 Some input visual scenes and the corresponding predicted saliency maps in SOD. 2

Figure 1-2 The differences between MSOD and instance-level RSR. . 5

Figure 2-1 Stacked representation of ground truth maps [81]. 36

Figure 2-2 Relative and absolute representations of ground-truth [81]. 40

Figure 3-1 The distribution of the proposed MSOD dataset. 61

Figure 3-2 Image examples in our proposed MSOD dataset. 61

Figure 3-3 Groundtruth examples in our proposed MSOD dataset. ... 62

Figure 3-4 The overall pipeline of our proposed approach, here shown using a VGG backbone. The red, orange, and green boxes capture saliency features, non-local features, and edge features respectively. Element-wise multiplication operates between each pair of ERB-DSNLB (edge and non-local features) and ERB-Conv (edge and saliency features). Our final prediction map is generated based on the fusion of 6 multi-scale saliency features in top-down pathway. 63

Figure 3-5 The architecture of a dual-space non-local block (DSNLB). C, H and W demonstrate the channel number, height and width of given feature map respectively and K = H × W. 65

Figure 3-6 The structure of a feature fusion gate. N, E, F and S demonstrate non-local feature, edge feature, saliency feature and the corresponding side output of bottom-up pathway respectively. 69

Figure 3-7 Feature visualization of non-local features, edge features and the refined features after feature fusion 70

Figure 3-8 Precision (vertical axis) recall (horizontal axis) curves on three popular salient object detection datasets and the proposed

MSOD dataset. The red solid line demonstrates our proposed method. 81
Figure 3-9 Qualitative comparisons with state-of-the-art approaches over some of the challenging images. The main object classes are statuette, chairs, human, bowling, and human (from top to bottom). 84
Figure 3-10 Different architectures of NLGM. All structures here are without FFG and ERGM. Element-wise addition operation is used at each stage to fuse different features. 89
Figure 4-1 An example of the mouse-contingent stimuli proposed in SALICON dataset [80]. The red circles indicate the movement of mouse cursor from one object to another. 94
Figure 4-2 Some common images in all three datasets. 101
Figure 4-3 Gaze data and velocity chart from one subject while looking at the image above. Here, 9 effective fixation events are captured after the threshold. 107
Figure 4-4 Examples of noise fixation points in gaze recording. 107
Figure 4-5 Several strategies applied to construct initial draft dataset. 111
Figure 4-6 Example 1: an image for participants to annotate in LabelMe software without crowded polygons. 113
Figure 4-7 Example 2: an image for participants to annotate in LabelMe software with crowded situation. 113
Figure 4-8 Examples of given images and the corresponding saliency ranking ground truth maps and MSOD ground truth maps. 116
Figure 4-9 An example of the annotations compiled in JSON file. Some JSON structure has been removed for clarity. 117
Figure 4-10 Examples of generated images in our dataset. 119
Figure 4-11 Statistics on the image quantity of different salient instance numbers in each image. 120

Figure 4-12 Statistics on the salient instances categories. 122
Figure 4-13 Statistics on average normalized saliency ranking in different categories. Blue bars indicate the data in our proposed dataset, while orange bars demonstrate the data in ASSR dataset. Note that ASSR dataset set a limit of salient instances to 5. 124
Figure 4-14 Average complexity of each category in our proposed dataset. 126
Figure 4-15 Statistics on foreground salient objects size ratio on three RSR dataset. 128
Figure 4-16 Background-contamination examples in IRSR and ASSR. 130
Figure 4-17 Statistics on instance location on three RSR datasets. .. 132
Figure 4-18 Statistics on global contrast and local contrast of each instance on three RSR datasets. 134
Figure 5-1 The process of Patch Merging. 147
Figure 5-2 Difference between traditional convolutional operation and the proposed convolutional operation on channel space. 157
Figure 6-1 Illustration of graph representation [101]. 162
Figure 6-2 An example graph. There are 5 vertices in this graph: B, C, M, E and F. 163
Figure 6-3 The attention mechanism in GAT. 165
Figure 6-4 Example of the limitations of GNNs. 167
Figure 6-5 The overall architecture of one QAGNet layer. Here, SSG, MSG and GRG demonstrate the Single Scale Graph, Multi Scale Graph and Global Representative Graph. 174
Figure 6-6 Illustration example of the tri-tiered nested style graph. 179
Figure 6-7 Qualitative comparison between our proposed method and other saliency ranking approaches on our proposed dataset. 190

Chapter 1 Introduction

1.1 Salient Object Detection

The human brain and visual system are equipped with the capacity to swiftly focus on significant areas within visual scenes. This particular capability, known as visual attention and visual saliency, has been a pivotal research topic in fields such as cognitive science, neuroscience, and psychology. It has also drawn considerable interest from computer vision researchers, as understanding this capability could aid them in identifying the most unique elements that can effectively represent an image.

Regarding visual saliency, two types of computational models have been established: Salient Object Detection (SOD) and Fixation Prediction (FP). Both models originated from the same community, but they serve different objectives. FP aims to highlight the points that human beings focus on when viewing a scene (e.g., Borji et al. [38][39]), while SOD aims to emphasize the most important salient object in an image. A strong correlation exists between the fixation points predicted in FP and the salient objects predicted in SOD. This is because both computational models often produce continuous-valued saliency maps, where pixels with higher values indicate the corresponding areas are likely to be more important and hence, attract more attention.

SOD typically involves two steps: 1) identifying the most noticeable object within a given image, and 2) segmenting this salient object in the image. As illustrated in Figure 1-1, the diagram includes four input images and their corresponding saliency maps generated by the SOD. For instance, in the top-left portion of this figure, an image including some buses is initially fed into the SOD model. Subsequently, the SOD model begins to detect the most significant salient objects in this image. Ultimately, the salient objects in this

image, the buses, are segmented as a binary saliency map, with pixels of higher values (shown as the white area) representing areas more likely to contain the salient object in the image.

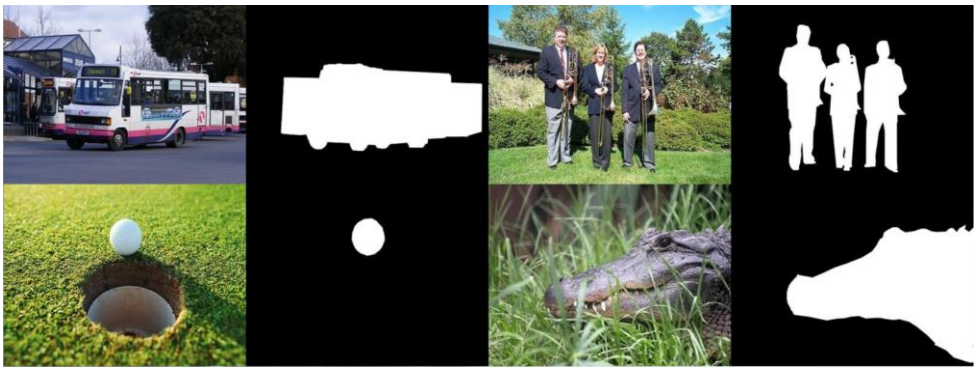


Figure 1-1 Some input visual scenes and the corresponding predicted saliency maps in SOD.

As SOD seeks to highlight the most visually striking or crucial objects within a scene, it plays an essential role in computer vision pipelines, SOD has been widely used in numerous object-level tasks across various domains. These include object recognition [45], object detection [46][47], image retrieval [48], image captioning [49][50], weakly supervised semantic segmentation[51][52], few-shot learning [131], and image cropping [53].

The origin of SOD can be traced back to the pioneering work carried out by Liu et al. [40] and Achanta et al. [41]. Drawing inspiration from earlier models designed to detect salient regions (such as [42] and [43]), they were the first to frame visual saliency issues as a binary segmentation problem. This has led to an increased interest in SOD, resulting in the development of numerous SOD models. Most early SOD models depended on low-level features like [3] and [4] or heuristic priors such as contrast [5] and background prior [6]. However, these early models, due to their reliance on hand-crafted features, faced difficulties in capturing high-level semantic information, thus limiting their robustness in tackling complex scenarios.

As deep learning technologies have progressed successfully in the field of computer vision, Convolutional Neural Networks (CNNs) [44] have

demonstrated their strong ability for visual feature extraction and representation. Consequently, there has been a growing emergence of deep learning-based SOD models since 2015. By leveraging multi-level and multi-scale features, CNNs are capable of accurately identifying the salient object without the need for any prior knowledge, such as information about the background.

Despite significant advancements, numerous open challenges still remain. One such problem lies in the nature of the datasets currently being utilized. Most of these datasets are predominantly populated by images with a single, typically centralized, object. Indeed, this conventional practice does reflect a common characteristic of human visual perception, where our attention is often immediately drawn to a central object. However, this focuses on centrally placed single object oversimplifies the vastly intricate process of human vision. We are capable of identifying not just one, but multiple salient objects distributed across a visually complex scene. Therefore, the simplicity of the current datasets potentially misrepresents the inherent complexity of human visual perception, making this an interesting and important area that requires further exploration. We explore this problem in Chapter 3. We curate a new dataset that is more challenging than previous SOD datasets. Rather than relying on the standard single-object format centered in the visual scenes, our dataset comprises images containing multiple salient objects scattered across a given visual scene. This is an important step towards more accurately mirroring the complex nature of human vision within the dataset, allowing us to delve deeper into the intricacies of visual perception.

To complement this new dataset, we also propose a novel architectural solution to the Multiple Salient Object Detection (MSOD) problem. By solving this problem, which is more complex than the traditional SOD problem, we are seeking to push the boundaries of what is possible in SOD, moving us closer to models that can mimic the depth and complexity of

human visual perception. We believe this to be a crucial component in advancing the field and bringing us closer to the goal of creating truly intelligent vision systems.

1.2 Relative Saliency Ranking

SOD aims to detect and segment the most visually prominent objects in a visual scene. Human beings have the ability to pay attention to multiple salient objects simultaneously [103] and this inspires us to explore the task of MSOD task in depth. However, simply detecting and segmenting multiple salient objects falls short in capturing the depth of human perceptual dynamics. It fails to reflect the inherent hierarchy of attention wherein not all salient objects command the same degree of focus. Without this relative ranking, the saliency representation remains incomplete, potentially limiting the efficacy of vision-based systems in applications that necessitate a deeper, more human-like understanding of visual priority.

Relative Saliency Ranking (RSR) is a new task with only few studies delving into its depths. It provides a hierarchical discernment by assigning rankings, reflecting the varying degrees of saliency among detected instances (see Figure 1-2). When humans observe a scene, they don't merely notice the salient objects, they intuitively prioritize their attention among multiple salient objects contingent on intuitive cues in the visual scene, e.g., instance scales and color, and their respective degrees of prominence.

Figure 1-2 The differences between MSOD and instance-level RSR.

Saliency ranking can benefit many down-stream tasks, such as image captioning [123][124], image cropping [125], video conversion [126] and autonomous driving [127]. In pioneering efforts, RSDNet [81] introduces the task of saliency ranking by assessing relative saliency values across pixels. However, its approach is limited to pixel-level saliency ranking without addressing object-level relative saliency. This limitation arises from its inherent design, which predicts saliency individually for each pixel, neglecting the ranking among distinct object instances. To attain object-level RSR with their approach, they need to leverage GT instance segmentation maps, which may not be feasible for real-world implementations. ASRNet [78] and IRSR [110] propose the instance-level RSR tasks, significantly boosting the development of RSR area. However, although remarkable process has been made, there still remain some open challenges.

From the perspective of datasets, both [78] and [110] have formulated instance-level RSR datasets by combining MS-COCO dataset [79] and SALICON dataset [80]. The former offers pre-defined polygons, while the latter captures mouse-trajectory-based fixations while participants moving

the mouse to explore the interested area in given blurry images. Although mouse-trajectory-based data in SALICON dataset can provide the mouse movement patterns, it resembles more of a human's exploration path based on their interests, which is not ideally suited for RSR. This limitation stems from the current assumption in SRD datasets that mouse movements are an adequate substitute for real eye-fixation data, which is typically gathered through gaze measurements in controlled settings. However, this assumption is problematic for several reasons. Firstly, mouse movements are controlled consciously, in contrast to a significant portion of eye movements, particularly saccadic movements, which are reflexive and thus represent different saliency aspects [129]. Secondly, the way the human brain processes mouse movements and eye fixation shifts likely occurs in distinct reference frames. Consequently, mouse pointing behaviors might also display biases and constraints due to the differences in these response mechanisms [130].

From the perspective of models, ASRNet models the saliency ranking task derived from human-beings' attention shifts. This is more like a scan-path prediction [122] rather than RSR. On the other hand, IRSR introduce the person prior information to predict the saliency ranking, which is bias against and unfair to the salient instances in other classes.

Among most of the saliency ranking methods [78] [81][112][113], they set a fixed number of output to predict limited saliency ranks, this is because the current datasets, ASSR and IRSR, set an arbitrary limit on the salient instances. However, setting a fixed number of outputs in saliency prediction models is a simplification that often fails to capture the real nuanced ways in which humans prioritize and attend to visual stimuli. Unlike machine models, our attention is not limited to a set number of items or regions in our visual field. Instead, it is continuously shifting and adapting based on our goals, experiences, and the context of the visual scenes.

Addressing these challenges, we employ an eye-tracker to create a

large-scale dataset for instance-level saliency ranking based on the fixation duration without setting a fixed number limit on the number of salient instances, which is introduced in Chapter 4. To the best of knowledge, this is the first and largest dataset created by real human fixations for RSR. This data collecting strategy integrates the naturally viewing patterns of human observers, offering a closer approximation to real-world perception compared to other datasets. Besides, a novel QAGNet framework is proposed, modeling multi-scale ranking-aware information cues in a tri-tiered nested style graph based on query-based transformers and this will be introduced in Chapter 5.

Understanding the RSR provides a deeper comprehension of the underlying principles of human vision. As computer vision continues to strive for human-like perception capabilities, studies in this domain could potentially bridge the gap between machine vision and the intricate processes of the human visual cortex. Thus, delving into the area of instance-level RSR is not just an academic pursuit; it holds potential real-world implications for the evolution of computer vision systems and the quest for AI that perceives the world as humans do.

1.3 Objective and Contribution

The primary objective of this research is to deepen the understanding and enhance the capabilities of SOD and RSR in computer vision. This is achieved through the introduction of novel datasets and computational models that more closely mirror human visual perception. The contributions of this research can be summarized as follows:

(1) Creation of a New MSOD Dataset: Addressing the limitations of existing SOD datasets, where the images are with a single, centrally placed object, this research introduces a new dataset. This dataset is distinct in that it comprises images containing multiple salient objects dispersed throughout

the visual scene. The introduction of this dataset marks a step towards more accurately replicating the complexity of human vision and understanding visual perception.

(2) Development of a Novel MSOD Model: In line with the new dataset, a novel model for the MSOD problem is proposed. This model aims to push the boundaries of traditional SOD by accommodating the detection of multiple salient objects in a single scene, thereby moving closer to replicating the depth and intricacy of human visual perception.

(3) Advancing RSR with a Large-Scale Dataset Using Human Fixations: Current RSR datasets are typically created based on the mouse-tracking data, which cannot reflect the real human visual system. This research proposes a large-scale RSR dataset based on real human fixations. This dataset does not impose a fixed number of salient instances, thereby offering a more natural and comprehensive understanding of human attention in visual scenes.

(4) Development of a Novel RSR Model: A novel framework based on transformer techniques and graph reasoning is proposed, which leverages the query features from a query-based transformer into a nested style graph. This framework is crucial for modeling the nuances of human attention and visual priority.

In summary, the research not only tries to address current challenges in SOD and RSR but also paves the way for future advancements in computer vision.

1.4 Thesis Structure

In this thesis, Chapter 2 provides background on both SOD and RSR. The task of SOD is similar to semantic segmentation, while the task of RSR typically involves instance segmentation. Therefore, the background of both semantic and instance segmentation is also introduced in Chapter 2. Chapter

3 introduces our contributions in MSOD. This includes the illustration of the proposed novel architecture and a curated dataset for testing competitive models' ability on MSOD task. An in-depth exploration of the newly created dataset for instance-level saliency ranking is presented in Chapter 4, containing the data collecting strategy, dataset structure and the statistics information on the proposed dataset. Chapter 5 introduces the exploration of transformer-based techniques. Chapter 6 demonstrates our proposed nested-style model, designed specifically for RSR. Finally, a conclusion is given in Chapter 7 illustrating the key findings and contributions of the works in this thesis.

1.5 Papers from This Thesis

1. Song, H., **Deng, B.**, Pound, M., Özcan, E., & Triguero, I. (2022). A fusion spatial attention approach for few-shot learning. *Information Fusion*, *81*, 187-202. [The content regarding SOD models in this thesis is applied to generate saliency maps, telling the feature extractor in this paper where to focus, which finally improves the performance of few-shot learning.]
2. **Deng, B.**, French, A. P., & Pound, M. P. (2023). Addressing multiple salient object detection via dual-space long-range dependencies. *Computer Vision and Image Understanding*, *235*, 103776. [This paper proposes a novel architecture for MSOD task. To evaluate competitive models' ability on MSOD, a new dataset only including multiple salient object images is curated. The detailed information of this work can be found in Chapter 3]
3. **Deng, B.**, Song S., French, A. P., Schluppeck D. & Pound, M. P. (2024). Advancing Saliency Ranking with Human Fixations: Dataset, Models and Benchmarks. [This paper is accepted by CVPR 2024, including the content introduced in Chapter 4 and 5]

Chapter 2 Background

2.1 Introduction

This thesis explores the design and training of deep neural networks for SOD and RSR. This chapter introduces the background of the SOD and RSR problem.

SOD, to some extent, can be defined as a special semantic segmentation problem, where semantic segmentation tries to segment all the objects into binary segmentation maps, while SOD only focuses on segmenting the visually striking objects. Therefore, not all segmented objects from semantic segmentation can be regarded as salient objects. In this chapter, Section 2.2 will first demonstrate the popular semantic segmentation architectures used in SOD and then, Section 2.3 will introduce the state-of-the-art models in SOD area.

In comparison, RSR task not only detects the salient objects, but also needs to differentiate the detected salient objects according to the saliency degree, which cannot be achieved directly through semantic segmentation ways. Instance segmentation is appropriate for detecting the distinct salient objects to be used for ranking reasoning. Therefore, before introducing the RSR works, instance segmentation background is firstly introduced in Section 2.4. Following that, current state-of-the-art RSR models will be described in Section 2.5.

2.2 Semantic Segmentation

Fully convolutional neural networks (FCNs) [2] and U-Net [3] have strongly boosted the development of SOD, inspiring a lot of researchers to build FCN-based and U-Net-based architectures for SOD with much better performance compared to previous models.

2.2.1 Fully Convolutional Neural Network

FCN is a classic deep learning architecture, which is designed for the semantic segmentation problem. Compared to traditional CNNs using fully connected layers for prediction, FCN utilized fully convolutional operations to make pixel-level dense predictions, being able to assign semantic labels for each pixel in an image.

The novel idea of FCN is to replace all the fully connected layers to convolutional operations. This operation makes the proposed method suitable for the input of different sizes and generating the same size output maps as the input image. The convolutional operations in traditional CNNs are used to extract the input image's local features, and the fully connected layers are utilized to map the extracted feature to the classification prediction in a fixed size. In comparison, FCN combines the two operations, making itself can not only capture the local feature but also keep the space information and finally achieve end-to-end pixel-level classification.

2.2.2 U-Net

U-Net [3] was proposed by Ronneberger et al., in 2015 for biomedical image segmentation. It is an FCN-based architecture. In U-Net, the researchers modified and extended the architecture of FCN, resulting in an architecture that can be trained in a small number of images but achieve accurate segmentation results.

The main idea of U-Net is to add a decoder similar to the encoder to make the whole architecture symmetrical. In the decoder of U-Net, there are also several convolutional operations, which generate multi-channel feature maps at different stages. This operation, to some extent, improves the feature representation. To better fuse the features in different levels, U-Net uses skip connections to combine the corresponding features in the encoder and

decoder. This operation makes the decoder get the information from different levels, therefore helping the model keep more detailed information.

The architecture of U-Net inspires researchers in the computer vision community and at the same time, promotes progress in deep learning architectures and information fusion strategies.

2.3 Salient Object Detection

SOD is a special semantic segmentation task, but only focuses on generating the binary segmentation map for the visually important objects. The popular semantic segmentation works have been introduced in Section 2.2, and here, the state-of-the-art SOD models are presented.

Most traditional approaches for SOD primarily depend on low-level features [3][4] or heuristic assumptions like color contrast [5] and background [6][7].

Early SOD methods based on deep learning mostly utilized multi-size image patches [8][9][10]. These approaches are delivering impressive results but are constrained by the absence of spatial context present in smaller image patches.

Since the introduction of fully convolutional networks (FCNs) by Long et al. in 2015, numerous efficient and successful end-to-end SOD architectures have emerged. In particular, U-shape-based architectures have gained substantial popularity.

Hou et al. 2017, 2019 [14][16] introduces short connections operations between deeper layers and shallower layers to merge high-level features and low-level features. Zhang et.al, 2018 [17] incorporates a gated pathway to facilitate bi-directional message passing and integration of multi-level features. Zhang et. al., 2018 [18] adopted multi-path recurrent connections and novel spatial attention modules for generating the saliency maps. Chen et al., 2018 [26] constructed a reverse attention block to highlight the non-

object areas. Qin et al., 2019 [19] implemented a bottom-up and top-down architecture to enhance the coarse saliency maps produced by the prediction network, creating boundary-aware saliency maps using a hybrid loss approach. Feng et al., 2019 [20] utilized global perceptron modules and attentive feedback modules for global saliency detection and to establish encoder-decoder communications respectively. Zhao et al., 2019 [21] employed edge features to direct the extraction of multi-scale features from a U-shaped structure, then fuse multiple side-outputs into a final saliency map. Liu et al., 2019 [22] research the influence of the pooling layers in the U-shaped architecture and propose a global guidance module to transmit the localization information to the top-down pathway. A feature aggregation module is also proposed to further enhance the fused features. Wu et al., 2019 [23] propose a cross-refinement unit for exchanging the information between edge features and saliency features. Zhao et al., 2020 [24] develop a gated dual branch network which includes a Fold-ASPP module to improve the localization of salient objects of various scales. Pang et al., [25] propose a transformation-interaction-fusion strategy to get multi-scale features efficiently and a consistency-enhanced loss to address the disparity issue between foreground and background.

Inspired by the U-shape architecture, more and more advanced SOD models with excellent performance are proposed. The classic state-of-the-art models that significantly influence the SOD area will be introduced below.

2.3.1 Deeply Supervised Salient Object Detection with Short Connections

This work (DSS) [1] proposes a method that combines multi-level features extracted from FCN to enhance the representation of each layer. This is achieved by introducing short connections to the skip-layer structure. The authors find that the output in deep layers of backbones such as VGG and

ResNet typically contains high-level semantic information. However, due to the information loss during down-sampling operations, saliency maps generated from these features tend to have irregular shapes, especially when dealing with complex and cluttered saliency maps. On the other hand, the shallower layers normally encode rich spatial knowledge, which can be used to highlight the boundaries of the generated saliency map. Based on these observations, their approach is to combine the high-level and low-level features by establishing short connections to the skip layers in order to improve SOD performance.

DSS extends the VGG architecture by adding an convolutional block and incorporates six side outputs (skip layers) to improve feature representations. These skip layers facilitate the gradual transmission of features from deeper layers to shallower layers, enabling the fusion of multi-level and multi-scale information through short connections.

In addition, DSS employs six side losses for each side output, as well as a final fusion loss, to enhance the quality of the generated saliency map by effectively combining information from different levels. While the binary cross-entropy loss is commonly used in SOD models, DSS employs these loss functions in conjunction with the proposed architecture and feature fusion mechanisms.

To summarize, this work, as a pioneering effort utilizing deep learning techniques for SOD, achieves top-tier results by merging multi-level and multi-scale features through short connections. In this setup, the deep layers' high-level features are transmitted to the lower layers to assist in identifying the salient object, whereas the low-level features from the shallower layers aid in refining the irregular predicted saliency maps.

2.3.2 A Bi-directional Message Passing Model for Salient Object Detection

In comparison to the earlier work introduced in Section 2.3.1, this study [17] utilizes a more effective method for fusing features produced from each side output via a bi-directional information passing strategy. Basically, the authors indicate that (1) the unweighted direct concatenation operation traditionally used to combine feature maps from each side output is not ideal, as not all features from each level are consistently beneficial, and 2) the erroneous information can sometimes undermine the performance of the SOD model, potentially leading to inaccurate saliency maps. Therefore, they propose a mechanism to filter out useless information.

Basically, the entire architecture consists of a backbone (commonly, backbones such as VGG [27] and ResNet [28] are used for feature extraction in the SOD area, and in this paper, VGG-16 is employed), a Multi-scale Context-aware Feature Extraction Module (MCFEM), and a Gated Bi-directional Message Passing Module (GBMPM). Once an image is fed into the backbone, the five side outputs are first introduced to the MCFEM module. Each block in the MCFEM applies four dilated convolutional layers [29] with a kernel size of 3x3 and dilation rates of 1, 3, 5, and 7 respectively, to expand the receptive field and detect multi-scale information.

Then, the four feature maps produced from varying dilated convolutional layers are concatenated to form multi-scale contextual features $F^c = \{f_i^c, i = 1,\dots,5\}$. These features are then inserted into the GBMPM module, eventually generating feature maps $H^3 = \{h_i^3, i = 1,\dots,5\}$. The GBMPM module is important in this model as it facilitates the exchange of semantic information in high-level features and spatial context information in low-level features. This module incorporates two directional connections: one starting from the first side output and ending at the last, and

another following the opposite direction. For instance, considering $h_i^0 = f_i^c$, the process of transferring information from the lower side output to the deeper side output can be represented as:

$$h_i^1 = Down(\emptyset(Conv(h_{i-1}^1; \theta_{i-1,i}^1))) + \emptyset(Conv(h_i^0; \theta_i^1) \qquad (2-1)$$

where $Conv(*; \theta)$, $Down()$ and $\emptyset()$ demonstrates the convolutional layer, downsampling layer and ReLU activation function respectively. Simultaneously, the final feature of the i^{th} side output is computed by:

$$h_i^3 = \emptyset(Conv(Cat(h_i^1, h_i^2); \theta_i^3)) \qquad (2-2)$$

where $Cat()$ represents the concatenation operation. From Eq. 2-2, h_i^3 is robust as it encompasses both high-level semantic information and low-level spatial information.

However, the features produced from each side output may not consistently contribute to the prediction of the saliency map. The redundancy of information can degrade the quality of the resulting saliency maps. Consequently, this paper introduces a gate function as the message is being passed in the GBMPM module, therefore changing Eq. 2-1 as follows:

$$h_i^1 = Down(G(h_{i-1}^0;\ \theta_{i-1,i}^{g1}) \otimes \emptyset(Conv(h_{i-1}^1; \theta_{i-1,i}^1))) + \emptyset(Conv(h_i^0; \theta_i^1)(2-3)$$

where $\otimes$ conducts an element-wise product operation and $G(*; \theta^g)$ is the gate function including a 3x3 convolutional layer followed by an element-wise sigmoid function. By implementing the gate function, only the beneficial information will be transferred within the GBMPM module.

Finally, the produced feature map h_i^3 and the predicted saliency map from a higher level S_{i+1} are merged to create improved saliency maps. This fusion process can be represented as:

$$S_i = \begin{cases} Conv(h_i^3; \theta_i^f) + Up(S_{i+1}), i < 5 \\ Conv(h_i^3; \theta_i^f), i = 5 \end{cases} \qquad (2-4)$$

where the $Conv(*; \theta_i^f)$ denotes 1x1 convolutional layers. By employing this method, the g saliency maps from deeper layers will be

progressively transmitted to the shallower layers.

In conclusion, this paper introduces a novel bi-directional message-passing method for SOD. By using the MCFEM and GBMPM modules, high-level and low-level features can mutually interact, and the other information potentially detrimental to the quality of the saliency map is filtered out. Compared to the work mentioned in Section 2.3.1, this work gives a better approach to fuse the information from different levels with advanced performance.

2.3.3 EGNet: Edge Guidance Network for Salient Object Detection

It is widely accepted that the shallower layers of the backbone normally contain low-level features such as edges and texture details, while deeper layers encompass high-level features, including semantic information. However, the effective utilization of both low-level and high-level features remains a challenge in the SOD field. The researchers in this study suggest that good edge detection significantly enhances the performance of both segmentation and localization tasks. Therefore, compared to previous models that directly merge low-level and high-level features, this work [21] specifically models the edge information in shallower layers and effectively leverages the high-level features in deeper layers, ultimately generating high-quality saliency maps.

Following the settings of DSS, this study adds another side path (Conv6-3) to the basic VGG structure to be the backbone. Simultaneously, since side output 1 is too close to the input, the feature generated at this location is disregarded. The features generated from the other side outputs can be represented as $C^{(2)}, C^{(3)}, C^{(4)}, C^{(5)}, C^{(6)}$. Among these, $C^{(2)}$ contains better edge information [30] and is therefore selected to model the edge data. To acquire more comprehensive context information, a U-Net

structure is employed to output multi-resolution features. Three convolutional layers and a ReLu layer (T) are added to each side output to generate robust features. Simultaneously, an additional convolutional layer (D) is utilized to transform the feature maps into one-channel prediction saliency maps.

The Non-Local Salient Edge Feature Extraction Module (NLSEFE) is constructed to model edge information. As previously mentioned, the lower layers (Conv2-2) carry richer edge details, while the higher layers (Conv6-3) include semantic information. To model saliency object edges effectively, high-level semantic information is also required. As a result, the features from Conv2-2 and Conv6-3 are combined, with the combined features $\bar{C}^{(2)}$ represented as:

$$\bar{C}^{(2)} = C^{(2)} + Up\left(ReLU\left(Trans\left(\hat{F}^{(6)};\theta\right)\right);C^{(2)}\right) \qquad (2-5)$$

where $Trans(*;\theta)$ represents the convolutional layers with parameter θ, designed to alter the number of channels. $Up()$ is employed to upsample the high-level features to match the size of $C^{(2)}$, while $\hat{F}^{(6)}$ denotes the enhanced features from the result of side output 6 (Conv6-3).

After obtaining the combined features $\bar{C}^{(2)}$, 3 convolutional layers are utilized to enhance the edge feature $\bar{C}^{(2)}$ and produce the final salient edge feature F_E. An additional salient edge supervision mechanism employing cross-entropy loss is designed to supervise the edge features.

In the One-to-One Guidance (OTOGM) module, the enhanced edge features F_E are integrated with each enhanced saliency feature $\hat{F}^{(i)}$ to better promote the saliency features when localizing and segmenting the salient object. This process can be represented as:

$$G^{(i)} = Up\left(\emptyset\left(Trans\left(\hat{F}^{(i)};\theta\right)\right);F_E\right) + F_E, i = 3,4,5,6 \qquad (2-6)$$

Then, similar to PSFEM, a sequence of convolutional layers T will further enhance the generated features and a converting layer D will make

the multi-channel feature map into a one-channel prediction map.

In contrast to other works, this study prioritizes modelling the saliency edge features, which is a highly efficient and effective method for preserving the boundaries of the generated saliency maps. By adopting this approach, the low-level edge features and high-level semantic features mutually enhance one another, ultimately producing high-quality saliency maps with clear boundaries.

2.3.4 PoolNet: A Simple Pooling-Based Design for Real-Time Salient Object Detection

This work [22] primarily explores the function of pooling layers in SOD models as utilizing the pooling operations would lead to more efficient but also effective models. Using the U-shaped architecture as a foundation, the authors introduce a global guidance module (GGM) that guides high-level semantic information to each phase of the top-down pathway. They also propose a feature aggregation module (FAM) to combine coarse-level semantic data with fine-level features in the top-down pathway. This paper pioneers the investigation of pooling-based models with the goal of enhancing SOD performance.

A recognized issue with the U-shape structure is the progressive dilution of high-level semantic information from the backbone as it passes through the top-down pathway. To address this information loss, the authors propose a global guidance module, which incorporates a modified version of the pyramid pooling module (PPM) [31]. The information generated by this module is conveyed to each stage of the top-down pathway through the global guidance module (GGM).

To be more specific, the GGM's PPM consists of four layers, with the first and last layers being an identity mapping layer and a global average pooling layer, respectively. For the intermediate layers, adaptive average

pooling layers are used to maintain the spatial dimensions of the output feature maps at 3x3 and 5x5.

With the application of global guiding flows, the features generated by the PPM are directly transferred to the feature maps at varying levels, thus mitigating the information loss from dilution in the top-down pathway.

Meanwhile, a novel feature aggregation module (FAM) is proposed in this work. The input feature map is initially transformed to different scales through the use of average pooling layers with varying downsampling rates. Then, these feature maps are processed by a convolutional layer, then upsampled and combined. This type of architecture presents two main benefits. First, it assists the model in reducing the aliasing effect of upsampling when passing the high-level semantic information, produced by the PPM, to the feature maps at each stage. This is particularly significant when the upsampling rate for high-level features is large, for instance, 8. Simultaneously, each sub-branch in the FAM observes the local context at different scales, contributing to the network's broader receptive field.

In addition, this study also conducts an experiment involving joint training with edge detection. Images from both the SOD dataset and the edge detection dataset are alternately used during training, effectively enhancing the boundaries of the salient object.

To summarize, this paper explores the capabilities of pooling layers in the field of SOD. Through the introduction of the global guidance module (GGM) and the feature aggregation module (FAM) to the U-shape architecture, this research is able to produce high-quality saliency maps with distinct boundaries.

2.3.5 Pyramid Feature Attention Network for Saliency Detection

The authors of this study [32] propose that different feature maps should

have distinct roles in the creation of the saliency map. To address this, they propose a new framework called the pyramid feature attention network (PFAN) for SOD. Within the proposed architecture, a context-aware pyramid feature extraction (CPFE) module is utilized to capture the context features of multi-scale high-level features. Simultaneously, the model uses channel-wise attention (CA) and spatial attention (SA) techniques on the CPFE feature maps and low-level feature maps respectively, enhancing its ability to detect salient objects.

It is acknowledged that each salient object has unique shape, scale, and position characteristics, therefore the direct usage of convolutional layers and pooling layers may not effectively manage this complex situation. Drawing inspiration from the feature extraction work of SIFT [33], the authors design a module capable of extracting features that vary in scale, shape, and location.

Specifically, the side outputs of conv 3-3, conv 4-3, and conv 5-3 from a basic VGG-16 backbone are selected as the high-level features, which are then fed into the CPFE. To ensure that the extracted features contain information from different scales, shapes, and locations, atrous convolutional layers [34] are employed with dilation rates of 1, 3, 5, and 7 to capture multi-level features. These features are then aggregated through cross-channel concatenation. After obtaining the three features of different scales, the two smaller ones are upsampled to match the size of the largest one, followed by another cross-channel concatenation operation to generate the final output of the CPFE module.

Many existing models merge features from different levels indiscriminately, leading to information redundancy issues, which may potentially affect the final performance of the saliency map. The attention mechanism, with its ability to select features, is a good choice for feature fusion.

The channel-wise attention technique is employed after the context-

aware pyramid feature extraction module to assign higher weights to channels that respond strongly to salient objects.

Regarding spatial attention, it is solely applied to low-level features. However, these features might contain certain details that negatively impact the final saliency map. Therefore, to guide the spatial attention towards the salient region, the high-level features are used to generate weights for the pixels in the salient region based on spatial attention. Subsequently, these weights are used to perform element-wise multiplication with the low-level features. Ultimately, the upsampled high-level features and low-level features are combined to produce the final saliency map.

In conclusion, this study proposes a context-aware pyramid feature extraction module that employs various atrous convolutional layers and a channel-wise attention mechanism to capture high-level semantic information. For the low-level features, the study applies a spatial attention mechanism to solve the negative effects caused by background noise and to make the model focus more on the salient region.

2.3.6 Stacked Cross Refinement Network for Edge-Aware Salient Object Detection

Most of the existing studies suggest that combining edges and saliency features can enhance the performance of SOD tasks, however, the presence of redundant and inaccurate edge features may compromise the quality of the generated saliency maps. Therefore, this paper [23] explores the relationships between the binary segmentation result and edge feature maps and proposes that the boundary region in an edge map is a subset of the object region in the segmentation map. Inspired by this notion, a new framework called the Stacked Cross Refinement Network (SCRN) has been introduced.

SOD, as a binary classification problem, the ground truth saliency map can be defined as follows:

$$M_s = \{M_s^p, p \in (0,1), p = 1, \ldots, N\} \quad (2-7)$$

In this formula, p and N represent a single pixel of an image and the total number of pixels in an image respectively. Similarly, the edge map of this image can be symbolized as M_e. For any given image, the white pixels of M_s indicate the salient object, while the white pixels of M_e emphasize the edges. This forms a logical relationship:

$$\begin{cases} M_s \wedge M_e = M_e \\ M_s \vee M_e = M_s \end{cases} \quad (2-8)$$

This logical relationship will be employed in this study to construct SOD models.

Specifically, this work is based on ResNet50, where four levels of features, denoted as $F = \{F^i, i = 1,2,3,4\}$, are extracted from the backbone. For each level, two 1x1 convolutional layers are employed to extract corresponding features with 32 channels for two different tasks. For these two tasks, $S = \{S^i, i = 1,2,3,4\}$ and $E = \{E^i, i = 1,2,3,4\}$ are used to denote the features for SOD and edge detection respectively.

Drawing from the interrelationships between edge maps and binary segmentation maps, the concept of stacked cross refinement units (CRUs) is introduced to enhance multi-level features. Specifically, the feature S_n^i and E_n^i (*nth* CRU and *ith* level) are calculated using the features S_{n-1}^i and E_{n-1}^i, which can be defined as follows:

$$S_n^i = S_{n-1}^i + f(S_{n-1}^i, E_{n-1}) \quad (2-9)$$

$$E_n^i = E_{n-1}^i + g(E_{n-1}^i, S_{n-1}) \quad (2-10)$$

For each level feature of one task, the corresponding level of the other task can be used to refine it. For example, E_{n-1}^i and S_{n-1}^i are utilized to refine each other. The multiplication operation can imitate the Boolean AND calculation when refining edge features based on binary segmentation features. The function g for this process can be defined as follows:

$$g = Conv\left(E_{n-1}^{i} \otimes S_{n-1}^{i}\right) \qquad (2-11)$$

where $\otimes$ denotes the element-wise multiplication and Conv stands for a 3x3 convolutional layer.

However, implementing the Boolean OR operation directly is challenging. As a result, an alternative method is employed to enhance the segmentation features, with the function f being defined as follows:

$$f = Conv\left(Cat(S_{n-1}^{i}, E_{n-1}^{i})\right) \qquad (2-12)$$

where Cat denotes the channel-wise concatenation operation.

The high-level features encompass semantic information, whereas the low-level features incorporate spatial information. As such, this paper proposes a 'set-to-point' style for better encoding of multi-level features. More precisely, this method refines each layer feature of one task based on all other level features from the other task. For instance, E_{n-1}^{i} will be refined by the four other level features in the segmentation task ($S_{n-1}^{k}, k = 1,\ldots,4$) and the function g can be expressed as:

$$g = Conv\left(E_{n-1}^{i} \otimes \prod_{k=1}^{4} Up(S_{n-1}^{k})\right) \qquad (2-13)$$

In this equation, Up represents the upsampling operation coupled with a 1x1 convolutional layer. On the other hand, the function f of the 'set-to-point' style can be defined as follows:

$$f = Conv\left(Cat(S_{n-1}^{i}, Cat_{k=1}^{4}[Up(E_{n-1)}^{k})]\right) \qquad (2-14)$$

In conclusion, this paper introduces a novel SOD framework that draws upon the interrelationship between segmentation maps and edge features. Thanks to the Cross Refinement Unit (CRU) module, multi-level features can share information between different tasks (segmentation and edge detection) to gradually refine each other. This ultimately leads to the generation of highly accurate saliency maps.

Although remarkable progress has been made, there still remain many

open challenges in SOD area. Existing SOD datasets contain many images with a single object, often centered in the middle of the image. This makes most of the SOD methods solve this problem mainly on single salient object and ignore the relationship information between different objects. Human observers may be drawn naturally to centered objects, but in complex scenes they can identify numerous salient objects distributed throughout a scene. Therefore, we explore the possibility of multiple salient objects problem in Chapter 3.

2.3.7 Summary of Popular SOD Models

We have discussed SOD methods in Section 2.3, and their strengths, potential drawbacks, and more have been summarized in the Table 2-1. This table provides a clear overview of the main characteristics, advantages, disadvantages, and the results achieved by each method in terms of SOD performance. Through this comprehensive analysis of these advanced methods, we gain a deeper understanding of the current research dynamics in the SOD field, as well as the effectiveness and limitations of various techniques in tackling complex saliency detection challenges.

Method	Key Features	Advantages	Potential Disadvantages	Results
Deeply Supervised Salient Object Detection with Short Connections (DSS)	Combines multi-level features from FCN; adds short connections and side outputs to VGG architecture.	Enhances layer representation; improves irregular saliency map shapes.	May have complexity due to multiple connections and layers.	Top-tier SOD results; refined saliency maps.
Bi-directional Message Passing Model for SOD	Utilizes bi-directional information passing; implements gated functions to filter information.	Effective fusion of multi-level features; filters out detrimental information.	Complexity in bi-directional processing and feature filtering.	Advanced SOD performance; more accurate saliency maps.
EGNet: Edge Guidance Network for SOD	Focuses on edge information; combines low-level and high-level features effectively.	Enhances segmentation and localization; preserves saliency map boundaries.	Specialized focus on edge features might limit general feature integration.	High-quality saliency maps with clear boundaries.
PoolNet: Pooling-Based Design for Real-Time SOD	Explores pooling layers in SOD; introduces global guidance and feature aggregation modules.	Efficient and effective model; distinct boundaries in saliency maps.	Pooling operations might lead to information loss.	High-quality saliency maps; efficient performance.
Pyramid Feature Attention Network for Saliency Detection (PFAN)	Implements context-aware pyramid feature extraction; utilizes channel-wise and spatial attention.	Captures multi-scale high-level features; focuses on salient regions effectively.	Complexity in managing multi-scale features and attention mechanisms.	High-quality saliency maps with focused salient regions.
Stacked Cross Refinement Network (SCRN) for Edge-Aware SOD	Focuses on interrelationship between edges and saliency features; introduces Cross Refinement Unit (CRU).	Refines multi-level features; accurate saliency maps with edge awareness.	Complexity in managing cross refinement between edge and segmentation features.	Highly accurate saliency maps with refined features.

Table 2-1 Summary of popular SOD models with main characteristics, advantages, potential disadvantages, and results.

2.4 Instance Segmentation

We have introduced the background works related to SOD in Section 2.3 and 2.4. From this section, the related works for RSR will be presented. As the RSR task can be regarded as an instance segmentation work, the popular instance segmentation methods will be firstly introduced in Section 2.4. Following this, the state-of-the-art RSR works will be presented in Section 2.5. The RSR problem is a new task, which normally utilizes the instance segmentation work first to generate salient instances and then learn the relationships between these instances to generate the instance-level RSR.

As illustrated in [91], there are four similar tasks in computer vision community, viz., images classification, object localization, semantic segmentation and instance segmentation. To be more precise, image classification is a task defined as a process of identifying the class of an object within the image or providing a list of object classes present in the image according to their classification scores. Contrastingly, the task of object detection/localization not only identifies the classes of objects within an image but also determines their locations. This location information is typically represented in the form of bounding boxes or centroids, providing spatial context to the classified objects in the image. On the other hand, semantic segmentation aims to achieve a more granular level of inference by assigning labels to each pixel in an image. Each pixel is labeled according to the object or region it belongs to. Building upon this, instance segmentation takes a step further by solving both object detection and semantic segmentation simultaneously. It not only identifies the class of each object in an image but also differentiates between individual instances and segments them.

Instance segmentation is a challenging task in computer vision that involves detecting objects and their boundaries. There are two main types of models used for this task: two-stage models and single-stage models.

Two-stage models first generate a set of region proposals that might contain an object. This is typically done using a Region Proposal Network (RPN). In the second stage, these proposals are classified into specific categories, and bounding boxes and masks are generated for each one. The two-stage process allows these models to be highly accurate, as the second stage can focus on a smaller number of high-quality proposals. However, this comes at the cost of computational efficiency, as the two-stage process can be slower than single-stage methods.

In Comparison, single-stage models aim to perform object detection and instance segmentation in one step. Instead of generating region proposals, these models directly predict the class and shape of each object in the image.

Single-stage models are typically faster than two-stage models, as they do not need a separate proposal generation stage. However, they may not be as accurate as two-stage models, especially for complex scenes with many overlapping objects.

Mask RCNN, as a well-known two stage instance segmentation model, will be firstly introduced.

2.4.1 Mask R-CNN

Mask R-CNN [88] is a region-based CNN that extends the capabilities of Faster R-CNN [90]. Faster R-CNN is an object detection model. It is designed to identify the presence of objects in an image and classify them, while also providing their location within the image in the form of bounding boxes. It is a two-stage method: the first stage, called a Region Proposal Network (RPN), proposes candidate object bounding boxes, and the second stage uses these proposals to classify the objects and refine their bounding boxes.

By incorporating a parallel branch for predicting object masks based on

Faster R-CNN, Mask R-CNN, with dual functionality, is able to perform object detection and instance segmentation simultaneously.

The architecture of Mask R-CNN is a two-stage framework. The first stage involves a Region Proposal Network (RPN), a fully convolutional network that scans the input image and generates a set of candidate object bounding boxes. This process involves sliding a small window across the input feature map, generating multiple anchors of fixed sizes and aspect ratios at each window position. The RPN then predicts whether these anchors contain an object and refines the bounding boxes accordingly.

The second stage of Mask R-CNN, known as RoI Align, takes these proposals and performs three parallel tasks: object classification, bounding box regression, and mask prediction. The RoI Align layer is a critical component of Mask R-CNN, designed to address the spatial misalignment issue caused by the RoI Pooling operation in Faster R-CNN. By using bilinear interpolation, RoI Align accurately maps the original pixel locations to the feature map, preserving the precise spatial locations and thereby improving the quality of the predicted masks.

For each proposed region, Mask R-CNN first extracts features using RoI Align. These features are then passed through three parallel fully connected layers. The classification layer predicts the object's class, the bounding box regression layer refines the object's bounding box, and the mask prediction layer generates a binary mask for the object. The mask prediction is performed in a pixel-to-pixel manner, allowing for the definition of the object at an instance level.

The training of Mask R-CNN is performed end-to-end, using a multi-task loss function that includes classification loss, bounding box regression loss, and mask loss. The mask loss is a pixel-level binary cross-entropy loss, which encourages the precise prediction of the object mask.

During inference, Mask R-CNN first generates proposals using the RPN. For each proposal, it performs classification, bounding box regression,

and mask prediction. Finally, it applies Non-Maximum Suppression (NMS) to remove overlapping detections, resulting in the final set of object detections and their associated instance masks.

With its excellent performance, Mask R-CNN has set a strong benchmark in the field of instance segmentation, inspiring numerous subsequent works.

2.4.2 CenterMask

The CenterMask [92] model is a novel anchor-free one stage approach to instance segmentation that combines the strengths of the Fully Convolutional One-Stage Object Detector (FCOS) [93] with a spatial attention-guided mask branch.

FCOS is motivated by FCN [2], utilizing the pixel-wise prediction for object detection. The whole architecture of FCOS includes three key components: backbone, feature pyramid and head. The novel idea in FOCS is the proposal of using center-ness, which is to improve the quality of bounding box predictions. Center-ness is a measure of how close a location is to the center of an object. The center-ness score is calculated for each location within the bounding box of an object. It is defined as the minimum of the four normalized distances (left, right, top, bottom) from the location to the boundaries of the bounding box. The distances are normalized by the corresponding side length of the bounding box. The center-ness score ranges from 0 to 1, with 1 indicating that the location is at the exact center of the bounding box. During training, the center-ness score is used as a weighting factor in the calculation of the localization loss. This means that locations closer to the center of an object have a larger impact on the localization loss. This encourages the model to predict more accurate bounding boxes, as it is penalized more heavily for inaccuracies near the center of an object. During inference, the center-ness score is multiplied with the classification score to

produce the final detection score. This helps to suppress the scores of bounding boxes that are not well localized, improving the overall quality of the object detections.

Based on FCOS, CenterMask propose to utilize a spatial attention-guided mask branch for the one-stage instance segmentation task. After the backbone, the feature maps are then passed to the FCOS detector and the spatial attention-guided mask branch in parallel. The FCOS detector is responsible for generating class and box predictions, while the spatial attention-guided mask branch predicts a segmentation mask for each detected bounding box using a spatial attention map.

The final instance segmentation is obtained by combining the class and box predictions from the FCOS detector with the mask predictions from the spatial attention-guided mask branch. This approach allows CenterMask to perform instance segmentation in real-time, making it a practical solution for applications that require fast inference speeds.

The CenterMask model achieves state-of-the-art performance on the COCO dataset, demonstrating its effectiveness as a solution for instance segmentation.

2.4.3 SOLO

SOLO [94], namely Segmenting Objects by Locations, introduces a unique concept of "instance categories", which assigns categories to each pixel within an instance based on the instance's location and size. This innovative approach allows instance segmentation to be done in a single-stage anchor-free framework.

Specifically, the input image is initially divided into a grid of cells SxS. Here, for a simple illustration, S is set to 5. Then the instance segmentation task is reformulated into two branches: a classification branch and a mask branch. The size of the classification branch is SSC, where C represents

the number of categories. The size of the mask branch is HWS^2, where S^2 represents the maximum number of predicted instances. The S^2 can be mapped to the original image from top to bottom and from left to right.

When the center of the target object falls into a certain cell, the corresponding position in the classification branch and the corresponding channel in the mask branch are responsible for predicting the object. For instance, if an instance is assigned to cell (i, j), then channel $k = i * S + j$ on the mask branch is responsible for predicting the mask of the target. Each cell belongs to a single instance.

Due to the mask branch predicting S^2 channels, if the grid cell is set too large, the output channel will become excessively large. Therefore, the paper proposes an improvement method called the Decoupled Head. Specifically, the mask branch is split into two directions: X and Y, each with S channels. The mask output is obtained by multiplying the two branches in an element-wise manner, reducing the prediction channels from S^2 to $2S$, with no significant loss in accuracy observed in the experiments. In this case, to obtain the mask predicted by the k grid cell, it only needs to extract the ith channel from the Y branch and the jth channel from the X branch, and perform an element-wise multiplication, where $k = i * S + j$.

In summary, SOLO model directly utilizes the mask prediction method to do the instance segmentation, which achieves state-of-the-art performance, strongly inspiring further development of one-stage anchor-free models.

2.4.4 BlendMask

This work [95] presents a novel approach to instance segmentation that combines the strengths of both top-down and bottom-up methods. The proposed method, BlendMask, leverages the advantages of top-down methods (such as Mask R-CNN) in terms of speed and simplicity, while also incorporating the fine-grained pixel-level information typically provided by

bottom-up methods, and finally proposing a single-stage anchor-free framework.

The architecture of BlendMask is built based on the FCOS object detector. The bottom module utilizes features either from the backbone or the Feature Pyramid Network (FPN) to predict a collection of base elements. A single convolution layer is appended to the detection towers, which concurrently generates attention masks with each bounding box prediction. For every predicted instance, the blender extracts the bases within its bounding box and linearly combine them in accordance with the learned attention maps.

The BlendMask framework consists of three main components: a backbone network, an attention mechanism, and a blending module. The backbone network is responsible for extracting feature maps from the input image. These feature maps are then fed into the attention mechanism, which generates a set of attention maps. Each attention map corresponds to a potential object instance in the image.

The attention mechanism is designed to focus on the spatial context of each instance, effectively providing a rough localization of the object. However, unlike traditional top-down methods, which typically predict a binary mask for each instance, the attention mechanism in BlendMask produces a continuous attention map. This allows for more flexibility and can better handle instances with complex or irregular shapes.

The blending module is the final component of the BlendMask framework. It takes the attention maps and the feature maps as input to produce the final instance masks. The blending module is designed to refine the rough localization provided by the attention mechanism, adding detailed pixel-level information to the instance masks. This is achieved by applying a blending operation to the attention maps and the feature maps, effectively fusing the coarse top-down information with the fine-grained bottom-up information.

One of the key advantages of BlendMask is its simplicity. Unlike many existing instance segmentation methods, which often involve complex multi-stage processes or require sophisticated post-processing steps, BlendMask is a single-stage method that can be trained end-to-end. Furthermore, despite its simplicity, BlendMask achieves competitive performance on the COCO benchmark, demonstrating its effectiveness.

In summary, BlendMask presents a novel approach to instance segmentation that effectively combines the strengths of top-down and bottom-up methods. By leveraging an attention mechanism and a blending module, BlendMask is able to produce high-quality instance masks that capture both the coarse spatial context of each instance and the fine-grained pixel-level details.

2.4.5 Summary of Popular Instance Segmentation Models

Table 2-2 provides a concise overview of the instance segmentation methods, highlighting the innovative aspects and the balance between efficiency and accuracy for each method. These methods represent strides in the field of computer vision, particularly in the challenging task of instance segmentation. They demonstrate the ongoing evolution of techniques aiming to optimize both speed and precision in processing and interpreting complex visual data.

Method	Key Features	Advantages	Potential Disadvantages	Results
Mask R-CNN	Extends Faster R-CNN; incorporates a parallel branch for object masks.	Highly accurate; able to perform detection and segmentation simultaneously.	Computationally less efficient due to two-stage process.	Strong benchmark in instance segmentation; precise object masks.
CenterMask	Anchor-free, one-stage approach; combines FCOS with a spatial attention-guided mask branch.	Real-time performance; efficient for applications requiring fast inference.	May not be as accurate as two-stage models in complex scenes.	State-of-the-art performance on COCO dataset; effective for real-time instance segmentation.
SOLO	Single-stage, anchor-free; uses "instance categories" for segmenting objects by location.	Simplifies instance segmentation process; reduces prediction channels.	Complexity in managing instance categories and segmentation.	State-of-the-art performance; inspires one-stage anchor-free model development.
BlendMask	Combines top-down and bottom-up methods in a single-stage anchor-free framework.	Simple and end-to-end training; captures both coarse and fine details.	May face challenges with extremely complex object shapes.	Competitive performance on COCO; effective fusion of detailed instance masks.

Table 2-2 Summary of popular instance segmentation models with key features, advantages, potential disadvantages, and results.

2.5 Relative Saliency Ranking

RSR is a new task with only few studies explore this area in depth. RSR includes not only detecting the salient objects, but also giving different salient objects ranking information indicating the degree of saliency. This affects how people view and interact with the surroundings. So, studying RSR is not just interesting but also important for researchers that want to imitate or understand human vision.

2.5.1 RSDNet

The pioneering work of Islam et al. [81] makes an initial attempt on RSR on pixel-level. The authors here observe an issue of a consensus due to the ill-posed nature of defining what universally constitutes a salient object. Multiple observers might have varying opinions on what they deem salient, leading to the challenge of ranking salient objects. Rather than focusing solely on the binary classification of salient vs. non-salient, it is pivotal to address the relative ranking among salient objects. This perspective derives from the observation that some objects are more likely to be judged as salient than others, highlighting the existence of a relative rank among them.

The paper introduces a novel deep learning solution that is based on a hierarchical representation of relative saliency, using the PASCAL-S [77] dataset. Note the PASCAL-S dataset includes ground-truth maps where different salient objects have different colors based on the degree of saliency.

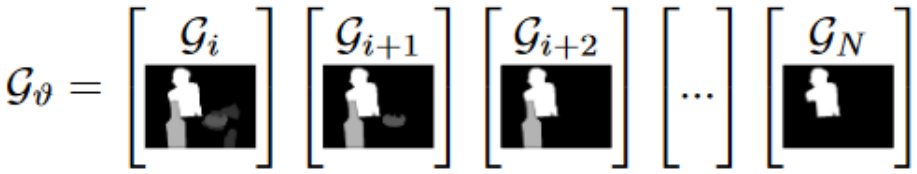


Figure 2-1 Stacked representation of ground truth maps [81].

In SOD or segmentation research, the ground-truth typically consists of numerical values that indicate the saliency level for each pixel. Historically, binary masks were created by thresholding, e.g., pixels are marked as white while pixels value more than 0.5, otherwise marked as black while pixels value less than 0.5. This ground-truth generation method does not reflect relative salience. Using such binary ground-truth masks is not ideal when the goal here is to model observer consensus. To address this limitation, this work introduces an approach to produce stacked ground-truth maps, each corresponding to a distinct saliency level determined by inter-observer consensus (see Figure 2-1). Given a base saliency map $\mathcal{G}_m$, a set $\mathcal{G}_\vartheta$

comprising N ground-truth maps $(\mathcal{G}_i, \mathcal{G}_{i+1}, \ldots, \mathcal{G}_N)$ can be derived. Each map $\mathcal{G}_i$ contains binary data indicating that a minimum of i observers perceived an object as salient, with this being represented on a pixel-by-pixel basis. Here, N represents the number of participants involved in annotating the salient objects. This stacked ground-truth, $\mathcal{G}_\vartheta$, offers enhanced differentiation between multiple salient objects and establishes a relative ranking that instructs the neural network to prioritize saliency levels. It's crucial to recognize the inherent hierarchy in the stacked ground-truth, where $\mathcal{G}_{i+1}$ is a subset of $\mathcal{G}_i$. This structure is vital because a format in which $\mathcal{G}_i = 1$ implies that i observers are in agreement could lead to zeroed layers in the ground-truth stack, and large changes to ground truth based on small differences in degree of agreement.

Regarding the overall architecture of RSDNet, the input image will be initially passed into an encoder (ResNet-101 for the best model) to generate a feature at $\frac{1}{8}$ scale. Then, an additional convolution layer with a 3 × 3 kernel and N channels is applied (where N represents the total number of individual observers participating in the labeling). This forms the Nested Relative Salience Stack (NRSS). Following this, they incorporate a Stacked Convolutional Module (SCM) to determine the preliminary saliency score of each pixel. The SCM is composed of three convolutional layers responsible for producing the targeted saliency map. The first convolutional layer has six channels with a 3 × 3 kernel. This is succeeded by two more convolutional layers: one with three channels and a 3 × 3 kernel, and another with a single channel and a 1 × 1 kernel. Each channel within the SCM is trained to assign a soft weight to every spatial position of the NRSS, facilitating the labeling of pixels based on their likelihood of being part of a salient object. This process can be expressed as:

$$\mathcal{S}_\vartheta^t = \mathcal{C}_{3\times3}(f_{\text{enc}}(I; \mathcal{W}); \Theta), \mathcal{S}_m^t = \partial(\mathcal{S}_\vartheta^t) \qquad (2-15)$$

where I denotes the input image, ($\mathcal{W}$, Θ) represent the parameters for

the convolution C, $\mathcal{S}_{\vartheta}^{t}$ is the initial Nested Relative Salience Stack (NRSS) for stage t. This stack captures varying saliency levels for every pixel, essentially predicting the likelihood of observers agreeing on an object's prominence. $\mathcal{S}_{m}^{t}$ stands for the initial saliency map, while ∂ symbolizes the Stacked Convolutional Module (SCM). The function $f_{\text{enc}}(.)$ produces the output feature map generated by the encoder network.

While the deeper layer of an encoder provides the most comprehensive feature set, solely using convolution and unpooling during decoding to retrieve lost details can diminish prediction accuracy. Therefore, here, the authors suggest a multi-stage fusion-based refinement network that, during decoding, combines initial coarse representation with finer feature maps at prior layers. This network comprises consecutive rank-aware refinement units that aim to restore lost spatial details during each refinement step, while also maintaining the relative ranking of salient objects. Every refinement stage uses the prior NRSS and earlier, sharper representations as its input, executing a series of operations to produce an enhanced NRSS. This aids in crafting a more detailed saliency map. It's vital to realize that enhancing the hierarchical NRSS means that the refinement process utilizes varying agreement levels from the SCMs to boost confidence in relative ranking and overall prominence. Lastly, the refined saliency maps produced by the SCMs are combined to form the final saliency map.

To integrate different features distinctly, a Rank-Aware Refinement Unit (RARU) is proposed. The RARU incorporates gate units, which regulate the information pass to reduce uncertainties related to figure-ground and salient objects. The first refinement unit $\mathcal{R}_{\vartheta}^{1}$ receives its input from the first NRSS $\mathcal{S}_{\vartheta}^{t}$ created by the feed-forward encoder. This refinement unit also uses the gated feature map $\mathcal{G}_{a}^{t}$, produced by the gate unit [106], as a second input. Following [106], they derive $\mathcal{G}_{a}^{t}$ by merging two successive feature maps f_{ξ}^{t} and f_{ξ}^{t+1} from the encoder. The prior $\mathcal{S}_{\vartheta}^{t}$ is upscaled to

twice its original size. A transformation function $\mathcal{T}_f$ — comprised of operations in sequence: convolution, batch normalization, and ReLU — is applied to the upscaled $\mathcal{S}_\vartheta^t$ and $\mathcal{G}_a^t$, resulting in the refined NRSS $\mathcal{S}_\vartheta^{t+1}$. Following this, the SCM module is used on top of $\mathcal{S}_\vartheta^{t+1}$, producing the refined saliency map $\mathcal{S}_m^{t+1}$. The $\mathcal{S}_\vartheta^{t+1}$ is then passed to the next stage's rank-aware refinement unit. These operations can be summarized as:

$$\mathcal{S}_\vartheta^{t+1} = w^b * \mathcal{T}_f\left(\mathcal{G}_a^t, u(\mathcal{S}_\vartheta^t)\right), S_m^{t+1} = w_s^b * \partial\left(\mathcal{S}_\vartheta^{t+1}\right) \qquad (2-16)$$

where u stands for the upsampling operation. The parameters for the transformation function $\mathcal{T}_f$ are represented by w^b, and w_s^b denotes the parameters for the SCM, which is indicated by ∂ in the equations. Note that $\mathcal{T}_f$ specifies a specific stage within the refinement procedure.

Furthermore, this paper also proposes other methods to deal with the stacked ground-truth maps. Specifically, the currently available dataset, PASCAL-S, offers data that facilitates the assignment of relative salience, based on consensus among several observers. Contrarily, in this work, they suggest ranking values in two distinct scenarios: Relative and Absolute.

In the Relative scenario, rank values are determined by the total number of instances in the mask and their rank score, denoted as $\mathbb{R}_\chi$, where χ is a specific instance. For instance, if a mask contains τ total instances, the range [0, 255] is divided by τ to generate the numerical rank value.

Conversely, in the Absolute scenario, rank values derive from the rank score set's percentile and are then adjusted to fit the range [50, 255], which equates to the gray-scale levels ranging from 20% to 100%. It will also produce a stacked representation of the ground truth.

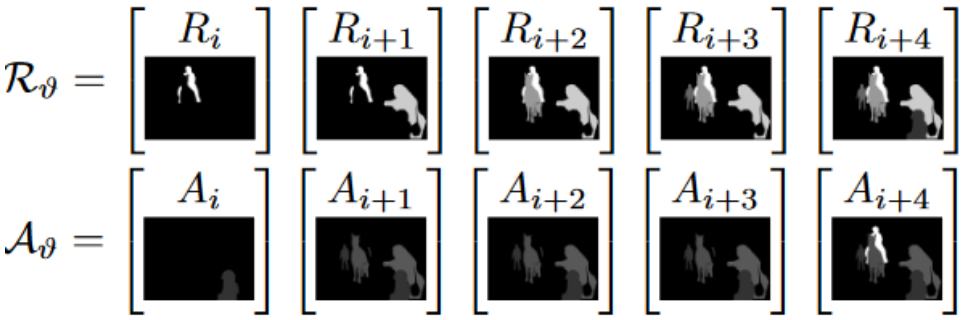


Figure 2-2 Relative and absolute representations of ground-truth [81].

As shown in Figure 2-2, it demonstrates the relative and absolute representations of ground-truth. For the relative scenario, the initial layer of the stack highlights the most prominent object, while the following layer denotes the top two salient objects, and so forth. As for the absolute scenario: the first layer represents less than 20% of the fixations and the next one represents less than 40% of the fixations. Therefore, in the relative scenario, each slice gets a single new instance. On the other hand, in the absolute scenario, several instances might be incorporated simultaneously to a slice if they share the same percentile rank.

This is the first work proposing the RSR problem. The methodology and findings detailed in this paper open new research avenues. Moreover, as the performance metrics on conventional SOD datasets seem to have reached their peak, alternative direction like rank order assignment is an interesting and promising area for future research. This paper in conjunction with the studies it builds on creates a robust groundwork for the future challenge of SOD.

2.5.2 ASRNet

This work [78] presents a novel approach to predict the saliency rank of objects in an image by inferring human attention shifts. This approach diverges from traditional methods that focus solely on identifying salient regions without considering the order of attention. Recognizing the sequence

in which humans shift their attention among objects provides a deeper understanding of human visual perception and can enhance the accuracy of saliency prediction models. To facilitate this research, the authors constructed a large-scale salient object ranking dataset. This dataset serves as a valuable resource for training and evaluating models designed to predict attention shift ranks.

The overall architecture of ASSR is composed of a backbone network, a Selective Attention Module (SAM), a Spatial Mask Module (SMM), and a network dedicated to classifying salient object rankings. Mask-RCNN is chosen as the bottom-up backbone, which offers object proposals using the FPN [107] and object segmentation from its segmentation branch. The SMM, in a bottom-up manner, draws out the low-level features of the proposed objects, while the SAM, operating in a top-down approach, focuses on advanced contextual attention features.

The SAM is constructed using the Scaled Dot-Product Attention mechanism [36], incorporating both image and object features. The pyramid feature P5, sourced from the backbone network, serves as the image feature. A 1 × 1 convolution followed by global average pooling is applied based on the pyramid features to generate high-level image feature. Prior to carry out the dot-product, the object and image attributes are projected into a 512-dimensional space. In this stage, each object's features are embedded into distinct feature vectors using a universally shared fully connected layer. Two distinct feature vectors are produced using separate FC layers, both of which use the pooled image features as their input. The newly formed feature sets, derived from the pooled image feature, are then repeated M times. The attention mechanism subsequently employs these embeddings to conduct dot product similarity comparisons between individual object features and the image features. A scaling factor is used and followed by a softmax activation to generate the attention score. The attention module calculates attention scores using multiple heads (specifically, 4 heads) simultaneously. The

results from the multiple attention heads are concatenated and then processed through an FC layer. Finally, a residual connection and an additional FC layer are incorporated to produce the module's output.

The authors propose that recognizing the relationships between the attributes of objects and the scene context is pivotal for selecting targets in complicated visual scene. For instance, small objects within a scene might not attract human gaze. Objects situated near the image's center might be perceived as more prominent, a phenomenon referred to the "center bias" principle [108][109]. Such insights drive the researchers' decision to incorporate low-level object features, such as size and position, to learn contextual features that can model the relationship between objects and the visual scene. By leveraging the bounding boxes of objects proposals, a spatial mask for each individual object is generated. These spatial masks contain the information of size and location of the object proposals related to the visual scene. (This map is generated using a binary mask, wherein pixels situated within a bounding box are assigned a value of 1, while all others are designated a value of 0.) These spatial masks are then passed through a set of convolutional layers to generated 64-D feature vector. Subsequently, the spatial features of each object are combined with their respective object features via a concatenation layer, followed by a fully connected layer. This process reduces the feature dimension to a fixed size of 512.

The authors' initial approach of modeling the detection of salient objects and the ranking sequence of attention shifts is to regard it as a classification task. In the configuration, only C = 5 ranks is considered. By incorporating an extra background category for non-salient objects, the classification encompasses 6 classes, which is the sum of 5 and 1. The prediction of saliency and rank is carried out through a classification network, which is composed of three convolutional layers followed by a single classification layer. At the inference stage, the saliency rank classification is

combined with object segmentation (sourced from the segmentation branch) to produce the final map indicating the rank of salient objects. However, a challenge arises with this classification-based approach: it does not guarantee that the identified salient objects will be allocated unique saliency ranks. To solve this limitation, softmax rank classification probabilities is applied in a scoring mechanism. For every object, the probability of its saliency rank is adopted as the initial score. This score is then enhanced by both addition and multiplication, based on a value corresponding to the predicted rank. This methodology draws inspiration from [76], which generates the saliency rank of objects based on the descending average pixel saliency value attributed to each object. Through this strategy, it can be ensured that the prediction of a distinct saliency rank for every object. In the final step, the top-5 saliency rank order of objects is considered, determined by their descending score values.

2.5.3 IRSR

Traditional SOD models are limited in their ability to distinguish the importance of different salient objects. Although recent studies, such as RSDNet [81] and ASRNet [78], have attempted to detect saliency ranking by assigning varying degrees of saliency to different objects, these models either fail to differentiate between object instances or place more emphasis on sequential attention shift order inference. This paper [110] introduces a new approach to address a practical problem that carries out simultaneous segmentation of salient instances and their relative saliency rank order. This model employs an enhanced Mask R-CNN for salient instance segmentation, followed by the addition of a saliency ranking branch to infer relative saliency.

The overall pipeline of IRSR is based on Mask R-CNN. Specifically, Mask R-CNN performs the instance segmentation and salient object

classification. An additional bottom-up pathway is introduced to enhance the transfer of low-level data to high-level feature maps, while also reducing the distance of information flow from the lower layers to the highest feature. Consequently, every multi-level feature can retrieve information from both lower and upper levels. Next, Region proposal networks (RPNs) are employed to produce salient object proposals. Each of the five-level features is passed to RPN to create proposals with a flexible scale. Following this, RoIAlign is utilized to extract RoI-specific feature, which is then directed to both a box head and a mask head. The former employs two fully connected layers for saliency classification and box regression for each proposal. In the case of the latter, the approach merges a convolution-deconvolution branch with a concurrent fully connected layer branch to achieve instance segmentation masks, benefitting the salient instance segmentation.

For the task of RSR, a novel graph reasoning module is constructed by integrating four distinct graphs. These graphs capture various aspects, including the interaction relation between instances, local contrast, global contrast, and a high-level semantic prior.

From the segmented salient instances and the saliency ranking feature map F, instance nodes $\{\mathbb{I}_i\}_{i=1}^{N}$, local context nodes $\{\mathbb{L}_i\}_{i=1}^{N}$, person prior nodes $\{\mathbb{P}_i\}_{i=1}^{N}$, and $M \times M$ global context nodes $\{\mathbb{G}_i\}_{i=1}^{M^2}$ are established. Subsequently, four graphs are constructed: an interaction relation graph $\mathcal{G}^r$, a local contrast graph $\mathcal{G}^l$, a global contrast graph $\mathcal{G}^g$, and a person prior graph $\mathcal{G}^p$. Within $\mathcal{G}^r$, every $\mathbb{I}_i$ is linked to others and itself. For $\mathcal{G}^l$ and $\mathcal{G}^p$, only $\mathbb{L}_i$ or $\mathbb{P}_i$ is linked to $\mathbb{I}_i$, respectively. In the context of $\mathcal{G}^g$, all $\{\mathbb{G}_j\}$ are connected to each $\mathbb{I}_i$. After several steps of graph reasoning, fully connected layers are used to generate saliency rank scores.

Additionally, a unique loss function is introduced to effectively train the saliency ranking branch. In the study by Islam et al. [81], the RSDNet is trained using the pixel-wise Euclidean loss between the predicted saliency

map and the ground truth (GT). Siris et al. [78] regards saliency ranking as a rank order classification challenge, employing a Softmax classifier combined with cross entropy loss to classify each instance into one of five rank orders. This paper focuses on predicting rank orders for images containing different numbers of salient instances. As a result, a ranking loss aligned with the GT rank order is introduced for training the saliency ranking branch. Drawing inspiration from [111], a pairwise ranking loss is utilized to promote higher saliency values for top-ranked instances and reduce values for those ranked lower. Specifically, for a training image with N instances, the GT ranks are represented as $\{r_1, r_2, \cdots, r_N\}$, with r_i ranging from 1 to N, and lower values signifying higher ranks. All possible instance pairs C_N^2 are extracted for training. For any given pair q, its two instances are ranked based on the GT ranks, meaning q is expressed as $q = \{q_1, q_2\}$ where both q_1 and q_2 fall between 1 and N, and r_{q_1} is less than r_{q_2}. With the predicted saliency scores of the two instances being s_{q_1} and s_{q_2}, the ranking loss is defined as:

$$L = \frac{1}{C_N^2} \sum_{q=1}^{C_N^2} \log\left(1 + \exp\left(-s_{q_1} + s_{q_2}\right)\right) \qquad (2-17)$$

However, the aforementioned loss treats every instance pair uniformly, neglecting the specific GT rank orders. This can be detrimental when optimizing the saliency scores of instances that are ranked extremely high or low. To address this, a dynamic loss weight, denoted as β, is introduced. This weight assigns greater values to pairs with significant rank differences and lesser values to those with similar ranks. This approach explicitly optimizes instances that have extreme ranks. Specifically, the enhanced ranking loss is represented as:

$$L = \sum_{q=1}^{C_N^2} \beta_q \log\left(1 + \exp\left(\left(-s_{q_1} + s_{q_2}\right)\right)\right) \qquad (2-18)$$

where β_p is determined based on the rank difference between q_1 and q_2, and $q_1 q_2$ can be normalized using:

$$\beta_q = \frac{\left(r_{q_1} - r_{q_2}\right)^{\gamma}}{\sum_{o=1}^{C_N^2} \left(r_{o_1} - r_{o_2}\right)^{\gamma}} \qquad (2-19)$$

Here, γ is a positive value. The larger its value, the more weight is given to pairs with significant rank differences.

This paper contributes a lot for the RSR problem, including the new state-of-the-art model and effective loss function, which strongly boost the development of RSR area.

2.5.4 SORNet

SORNet [112] is the first paper utilizing the Transformer technologies for RSR. Note the Transformer techniques will be reviewed and explored in Chapter 5.

Regarding the overall architecture of SORNet, a CNN based backbone is firstly utilized for feature extraction. It takes a raw image as input and outputs a feature map. Before the ROI pooling operation (which extracts object-level features from each proposal), the X and Y coordinates are combined with the feature map to form: [*FeatureMap; PositionMap*]. Then, the detection branch and SOR branch are parallel. The detection branch employs standard detection techniques, such as Mask RCNN and CenterMask. Its main objective is to identify objects and predict their positions, classifications, and associated masks. Notably, this branch doesn't utilize the positional data of each proposal. SOR branch is specifically designed to rank each proposal based on visual saliency. The SOR branch focuses on ordering proposals instead of merely identifying them. The key idea in this branch is the proposal of Position-Preserved Attention (PPA) module, which consists of two stages: position embedding and feature

interaction. In the initial stage, both semantic and positional data are integrated to produce visual tokens. These tokens are subsequently processed in the feature interaction phase, resulting in contextualized descriptions for each proposal. The process is followed by a fully connected layer that predicts the ranking order for each proposal. The total loss is calculated based on the detection loss and ranking loss, where detection loss includes the box loss, classification loss and mask loss, while the ranking loss is the cross-entropy loss between the ground truth ranking and predicted ranking order.

The key idea of this paper is the PPA module, which contains 2 stages: position embedding stage and feature interaction stage. The PPA module receives the feature representations of proposals with their positions as input. Specifically, the input dimensionality is $N \times 14 \times 14 \times (256 + 2)$, where N stands for the number of proposals, 14 represents the ROI pooling size and the channel number for the feature map and positional indices are 256 and 2, respectively. For a given proposal, denoted by its i-th bounding box ($bbox_i$), the post-ROI pooling feature is determined by $[\ fea_i; pos_i\ ]$ = RoIPooling([*FeatureMap; PositionMap*], $bbox_i$). The PPA module then produces contextualized representations for each proposal of dimension $N \times 1024$.

In the position embedding stage, the main goal is to integrate semantic features with the positional information of each proposal. This integration contains several steps. Initially, the feature map is divided into semantic and positional components. Following this, a convolution layer with a ReLU activation function is applied to the positional component to extract low-level features: $pos\_fea_i = \text{Conv}\ (pos_i)$. Subsequently, this newly derived feature is concatenated with the original positional data, resulting in a position embedding, expressed as $pos\_embeding_i = [pos_i;\ pos\_fea_i]$. The subsequent steps involve combining this embedding with the semantic

feature and processing the combined entity through a series of four convolution layers, formulated as $f_{ea_i} = Convs\left(\left[fea_i; pos\_embeding_i\right]\right)$. Then, a flatten operation is applied, followed by two fully connected layers to convert the feature into a one-dimensional vector encompassing 1024 channels, which is the visual token.

In feature interaction stage, the goal is to utilize the information between different proposals. To achieve this, the self-attention mechanism of the Transformer's encoder is leveraged. The methodology strictly aligns with the conventional Transformer encoder structure. Within this architecture, multi-head self-attention modules and feed-forward neural network (FFNN) units are applied.

In summary, this method is the first one using the Transformer technologies, achieving state-of-the-art performance in RSR task.

2.5.5 OCOR

This work [113] utilizes the query-based object detection models, such as QueryInst [114], to do the saliency ranking task.

Specifically, given an input image, the query-based object detection technique is utilized to extract global context features. Subsequently, a collection of trainable salient object proposals (e.g., box and object queries that represent object positions and detailed object attributes) are employed to predict the final saliency rankings. The Saliency Rank Learning process includes: (1) a Selective Object Saliency (SOS) module that refines object-level semantic details, (2) an Object Context Object Relation (OCOR) module that comprehends interactions between objects and their respective contexts in a bi-directional manner, and (3) ranking and mask heads that determine object-specific saliency rankings, building on the enhanced features from the SOS and OCOR modules.

In SOS module, global covariance pooling [115][116] is firstly used to

capture object features and understand their relationships with both local and global contexts. Following this, a set of dynamic rectifying functions are learned to adjust channel attentions based on the high-order feature statistics derived from global covariance pooling. As a result, they collaboratively extract detailed object data to develop object representations.

In OCOR module, the spatial attention mechanisms inherent in the human visual system with the goal of learning region prioritization is modeled. To achieve this, the object-context relationship is encoded using the enriched object representation from the SOS module. Then, a bi-directional object-context-object relationship is constructed to simulate the way humans look at visual scenes.

This model achieves state-of-the-art performance with high SOR score. However, there is a limitation to this approach. When objects with identical functions are present in a scene with limited context, the model might not determine the correct saliency ranking accurately.

2.5.6 Summary of RSR Models

Table 2-3 encapsulates the innovative strides each method has made in the field of RSR. It highlights their unique approaches to understanding and replicating human visual perception, focusing on not just identifying but also ranking the saliency of objects in complex visual scenes.

Method	Key Features	Advantages	Potential Disadvantages	Results
RSDNet	Hierarchical representation of saliency; stacked ground-truth maps based on observer consensus.	First work to propose RSR task.	Pixel-level saliency ranking is inpractical..	Pioneering in RSR; provides a foundation for further research.
ASRNet	Predicts saliency rank by inferring human attention shifts; large-scale salient object ranking dataset.	Combines low-level and high-level features; unique in considering attention shift order.	May struggle with accurately ranking objects in complex scenes with many salient items.	Innovative in predicting attention shift ranks; enhances accuracy of saliency prediction models.
IRSR	Enhanced Mask R-CNN for segmentation; graph reasoning module for saliency ranking.	Simultaneous segmentation and ranking; captures various aspects like interaction and contrast.	Using person prior information is unfair for other classes.	Effective in differentiating importance among salient objects; introduces a novel ranking approach.
SORNet	Uses Transformer technology; combines semantic and positional information for ranking.	First to apply Transformer tech in RSR; sophisticated integration of various data types.	Complexity in managing Transformer architecture and data integration.	State-of-the-art performance in RSR; innovative use of Transformer technology.
OCOR	Query-based object detection; focuses on refining object-level semantic details and context.	Advanced object-context relationship understanding; high saliency rank score achievement.	May not accurately rank objects with identical functions in limited contexts.	Good performance in saliency ranking; advancement in understanding object-context relations.

Table 2-3 Summary of popular RSR models with key features, advantages, potential disadvantages, and results.

2.6 Conclusion

In this chapter, the background of both SOD and RSR is illustrated. Existing SOD datasets contain many images with a single object, often centered in the middle of the image. Human observers may be drawn naturally to centered objects, but in complex scenes they can identify numerous salient

objects distributed throughout a scene. Therefore, the possibility of multiple salient object task is explored. RSR is a new area with only few methods and datasets. Current RSR datasets are constructed based on mouse-trajectory-based fixations, which cannot reflect the real human visual systems. To address this problem, we create a large-scale instance-level saliency ranking dataset as well as novel model for benchmarking. From the next chapter, the research gaps and the corresponding proposed solutions in both SOD and RSR will be introduced in detail.

Chapter 3 Multiple Salient Object Detection

3.1 Introduction

Chapter 2 provided an overview of the relevant work in both SOD and RSR. There has been much progress in these areas in recent years, but techniques continue to be trained and tested on specific and limited datasets often containing quite simple scenes, with only a few salient objects. The performance of these methods on more complex scenes comprising many salient objects is untested. In this chapter we then present a new approach to SOD based on a U-shape architecture, the key focus of this work is the use of dual-space non-local blocks to provide improved performance where objects are spread throughout a scene by better considering long-range dependencies between image features. To evaluate this work, and the performance of existing methods on complex scenes, we curate a new dataset drawn from multiple existing SOD datasets. This dataset focuses exclusively on scenes containing numerous salient objects, specifically three or more. Compared to the state-of-the-art methods, we show that our approach offers higher performance on both the existing datasets and the new curated dataset.

In section 3.2, the research gaps from SOD to MSOD is first illustrated. Then, the background related to our proposed method is introduced in section 3.3. Our proposed dataset is introduced in section 3.4. Section 3.5 demonstrates the architecture of our proposed method, while section 3.6 shows the experiments and results of our proposed model in detail. The content in this chapter has been accepted by the journal of Computer Vision and Image Understanding [103].

3.2 Background

In this section, the background works that highly inspire our proposed method for MSOD are introduced.

3.2.1 Non-local Neural Networks

Long-range dependencies are crucial in deep neural networks. Normally, when it comes to sequential data, recurrent operations are typically used to address long-range dependency issues. On the other hand, for image data, stacked convolutional layers with large receptive fields are commonly employed to model long-distance dependencies.

The work in [35] propose a straightforward and efficient non-local operation to capture long-range dependencies in deep neural networks. This operation calculates the response of a particular position by computing the weighted sum of all other pixels in the input image.

In this work, a non-local operation can be defined as follows:

$$y_i = \frac{1}{C(x)} \sum_{\forall j} f(x_i, x_j) g(x_j) \qquad (3-1)$$

where i indicates the index of the output pixel and j denotes the index of all the other possible pixels. x and y are the input image and the output image respectively. Meanwhile, f calculates a scalar value that represents the relationship between the output pixel at index i and all other pixels at index j, while the function g computes a representation of the input image at the position j. Finally, the result is normalized by factor $C(x)$. Compared to a traditional convolutional layer operation, which only considers the sum within a local neighborhood (e.g., $i - 1 \leq j \ll i + 1$), this proposed non-local operation is capable of effectively capturing long-range dependencies and global information.

For simplicity, this work only considers g as a linear function: $g(x_j) = W_g x_j$. In this function, W_g represents a weight matrix that is learned from a 1x1 convolutional layer. Then, after exploring several choices for function f, it was found that the dot product yields the best performance. The dot product function can be defined as follows:

$$f(x_i, x_j) = \theta(x_i)^T \emptyset(x_j) \qquad (3-2)$$

In this function, the normalization factor can be regarded as $C(x) = N$, where N denotes total number of pixels in the input image x.

In comparison, an alternative form of the function f can be computed using the Gaussian function, which can be expressed as:

$$f(x_i, x_j) = e^{\theta(x_i)^T \emptyset(x_j)} \qquad (3-3)$$

where $\theta(x_i) = W_\theta x_i$ and $\emptyset(x_j) = W_\emptyset x_j$ are embeddings. This version resembles the self-attention module [36], while the $\frac{1}{C(x)} f(x_i, x_j)$ corresponds to the softmax operation. Specifically, $y = softmax\ (x^T W_\theta^T W_\emptyset x) g(x)$.

This work embeds the Eq. (3-1) within a non-local block, which can be defined as follows:

$$z_i = W_z y_i + x_i \qquad (3-4)$$

where y_i has been given in Eq. (3-1) and $+x_i$ demonstrates a residual connection.

In conclusion, this research introduces a non-local operation that can be integrated into various deep learning architectures. The non-local block proposed in this work effectively captures long-range dependencies by computing relationships between any two pixels in an image. By incorporating global information, it enhances the performance of baseline models across different tasks.

3.2.2 Dual Attention Network for Scene Segmentation

This study [37] explores into the non-local operations with self-attention mechanisms for scene segmentation. In detail, it argues that prior research only employed multi-scale feature fusion mechanisms and convolutional operation for feature capture, which can only deal with the local receptive fields. To solve the problem, the study introduces two kinds of attention modules using the self-attention mechanism to capture the global contextual information, which are Position Attention Module and the Channel Attention Module.

The whole process of the Dual Attention Network is demonstrated. Specifically, a pre-trained residual network with a dilated operation serves as the backbone for this architecture. The architecture conducts down-sampling operations within the backbone and utilizes dilated convolutions in the residual network's final two blocks. The backbone's output is initially processed by two convolutional layers for dimension reduction, and subsequently, it is fed into the newly proposed Position Attention Module and Channel Attention Module. Ultimately, the outputs derived from these two modules are combined to achieve improved feature representations.

The position attention module is designed on the spatial space based on self-attention mechanism. For a provided feature map, A, with dimensions $A \in R^{C\times H\times W}$, A will be firstly fed into 3 convolutional layers, resulting in three feature maps $\{B, C, D\} \in R^{CxHxW}$. Then, these maps are then reshaped to $\mathrm{R}^{\mathrm{C}\times\mathrm{N}}$, where N denotes the number of pixels in a channel, e.g., $N = H \times W$. After that, a matrix multiplication operation is performed between the transpose of B and C. This is then followed by a *softmax* layer to obtain the spatial attention map $S \in R^{N\times N}$, which can be defined as follows:

$$s_{ji} = \frac{exp(B_i \cdot C_j)}{\sum_{i=1}^{N} exp(B_i \cdot C_j)} \quad (3-5)$$

In this equation, s_{ji} here indicates the i^{th} position's response on j^{th} position. In this case, when the similarity between two positions is high, there will be a greater correlation observed.

Simultaneously, a matrix multiplication is conducted between the transpose of S and D. The result of this operation is reshaped back to R^{CxHxW}. Eventually, a weight parameter α and a residual operation are employed to generate the final output $E \in R^{CxHxW}$:

$$E_j = \alpha \sum_{i=1}^{N} (s_{ji} D_i) + A_j \quad (3-6)$$

As inferred from the Eq. 3-6, each position value in E is a weighted summation of all the features. This illustrates the Position Attention Module's capability to capture global spatial information.

To explore the interdependencies amongst various channel maps, this paper introduces a novel Channel Attention Module.

Compared to the Position Attention Module, Channel Attention Module computes the attention map X, with dimensions $X \in \mathrm{R}^{\mathrm{C}\times\mathrm{C}}$, without the need of convolutional layers. Initially, A is reshaped into $R^{C\times N}$. Following this, a matrix multiplication operation is conducted between A and its transpose. A *softmax* layer is then applied, resulting in the generation of the channel attention map. This process can be mathematically represented as:

$$x_{ji} = \frac{exp(A_i \cdot A_j)}{\sum_{i=1}^{C} exp(A_i \cdot A_j)} \quad (3-7)$$

where x_{ji} demonstrates the i^{th} channel's response on j^{th} channel. Then, a matrix multiplication operation is carried out between the transpose of X and A, followed by a reshape operation. Similar to the Position Attention Module, a parameter β and an element-wise addition operation are

used to derive the final output E with dimensions$\mathrm{R}^{\mathrm{CxHxW}}$:

$$E_j = \beta \sum_{i=1}^{C} \left(x_{ji} A_i\right) + A_j \qquad (3-8)$$

In conclusion, this study introduces an innovative structure, termed the Dual Attention Network, designed specifically for image segmentation. Two novel modules, namely the Position Attention Module and the Channel Attention Module, are proposed with the self-attention mechanism. The architecture is special designed to capture the global dependencies in both spatial and channel space, thereby facilitating the improved generation of segmentation maps.

When compared to traditional image segmentation works, where the model generates segmentation maps with several objects from various classes labeled, SOD also produces segmentation maps but only highlights the salient object area. Therefore, there are many similarities between the semantic segmentation and SOD areas. These similarities greatly inspire my investigation into the role of global dependencies information in the SOD area.

3.2.3 Discussion

Regarding MSOD task, the given visual scenes usually not only contain a single salient object. Traditional popular SOD methods typically generate results with missing salient objects in the complicated visual scenarios with multiple salient objects. Inspired by the utilization of non-local information introduced in Section 3.2.1 and Section 3.2.2, the following chapters will investigate the use of non-local information for MSOD task. Compared to convolutional operation with limited receptive field, non-local information helps to give a global receptive field, reducing the risks of missing salient objects.

3.3 Research Gaps - From Salient Object Detection to Multiple Salient Object Detection

Although remarkable progress has been made, there still remain many open challenges. Existing SOD datasets contain many images with a single object, often centered in the middle of the image. Human observers may be drawn naturally to centered objects, but in complex scenes they can identify numerous salient objects distributed throughout a scene. On the other hand, many existing saliency techniques are based on traditional U-shaped networks that only involve convolutional operations processing local neighborhoods. The size of the receptive field is critical in locating and segmenting salient objects across the image. Larger kernels aid in these segmentation tasks, but the experimental receptive fields are usually smaller than the ones in theory [22][32][54]. This is likely to limit the performance of SOD networks, especially when objects are spatially separated. Long-range dependencies have been proven to play a crucial role in various classification tasks [35], and this is also applicable for pixel-level segmentation tasks like SOD. Current methods fail to leverage long-range pixel-wise or channel-wise relationships among features in an image, resulting in a reduced capability to address the issue of multiple salient objects.

Recent state-of-the-art SOD approaches tackle salient object detection by refining and combining multi-level features into feature representations [16][17][23][22][32], incorporating additional losses into frameworks to provide structural information [20][19], or applying attention mechanisms to filter out redundant information and focus on valuable features [18][26][20][23].

However, existing SOD methodologies rarely take into account long-range dependencies, which involve information sharing across spatially

distant pixels or between channel space feature maps. Among the few that do, Li et al., 2020 [55] and Sun et al.,2019 [56] deployed self-attention mechanisms to capture spatial long-range contexts. Zhou et al., 2020 [57] incorporated a multi-type of self-attention to capture pixel-level relationships for saliency detection in degraded images. Liu et al., 2020 [58] devised a self-mutual attention to seize long-range contextual dependencies in RGB-D. Despite these efforts, none have utilized channel-wise dependencies as we do here, and there is no ideal solution for MSOD. This has previously been hard to examine; existing public datasets contain some multi-object instances, but the frequency of these varies substantially. We have curated a dataset specifically for this purpose, allowing us to concentrate on this issue.

In Chapter 3, we propose a novel architecture for MSOD that considers long-range dependencies in both spatial and channel space. Drawing inspiration from existing work [37], we propose a non-local guidance module (NLGM), comprised of several dual-space non-local blocks (DSNLBs) to capture pixel-wise and channel-wise relationships. Features at each location are aggregated by a weighted sum of all features in spatial space, while each channel map is updated by a weighted integration of all interconnected channel maps. Different from previous work [37], we stack several DSNLBs to progressively capture non-local features. These non-local features and bottom-up convolutional features are fused in the decoder via feature fusion gates that manage the passage of information to the next stage of the decoder. This includes salient edge supervision to further improve the quality of the saliency maps. We demonstrate the improved performance of our network on various datasets, focusing also on MSOD problems by evaluating it on a dataset composed solely of complex multi-saliency images.

3.4 Proposed Dataset

Current popular SOD datasets include DUT-OMRON (Yang et al., 2013 [7]), HKU-IS (Li and Yu, 2015 [61]), DUTS (Wang et al., 2017 [59]), ECSSD (Yan et al., 2013 [60]), SOD (Movahedi and Elder, 2010 [62]). Specifically, the DUT-OMRON dataset consists of 5168 high-quality images featuring one or more salient objects and complex backgrounds. The HKU-IS contains 4447 challenging images, some of which have several disconnected salient objects. The DUTS dataset is the largest SOD benchmark, encompassing 10553 training images and 5019 testing images with various scales and locations. The ECSSD dataset includes 1000 images with semantically meaningful structures. The SOD dataset includes 300 challenging images.

Most of the images in these datasets only feature a single salient object. Although scenes with multiple salient objects are found in each dataset, their frequency varies from a minimum of 9.8% in ECSSD to 50.3% in HKU-IS, with an average of 28% across all datasets. In these multi-object images, the majority contain only two objects.

To specifically evaluate the performance of various state-of-the-art SOD models in MSOD task, we have curated a new dataset, namely MSOD. It consists of the most challenging multi-object scenes from the five existing datasets. To better evaluate our and other methods' effectiveness on multi-object images, we only include scenes with three or more salient objects. The complete MSOD dataset encompasses 300 test images including 1342 objects in total. The number of objects per image ranges from 3 to 19, as shown in Figure 3-1.

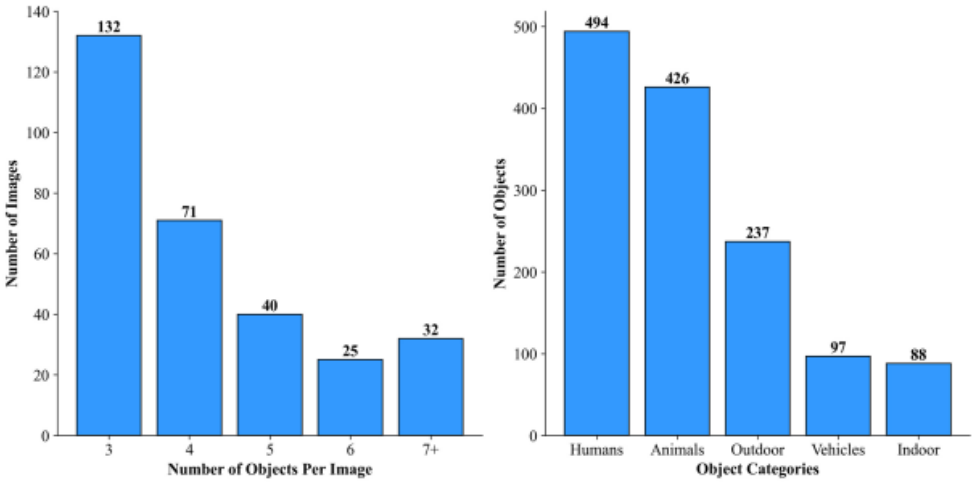


Figure 3-1 The distribution of the proposed MSOD dataset.

The dataset includes a diverse range of object classes and a varying number of these objects spread across each image.

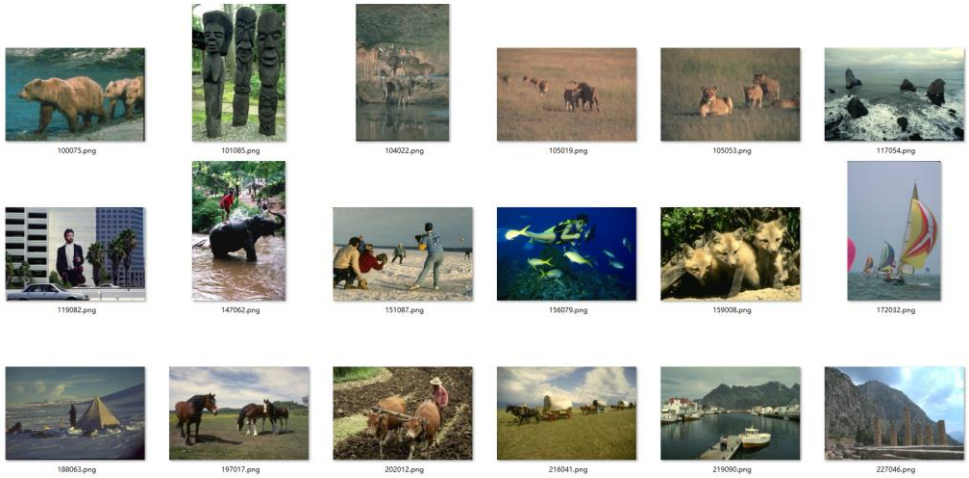


Figure 3-2 Image examples in our proposed MSOD dataset.

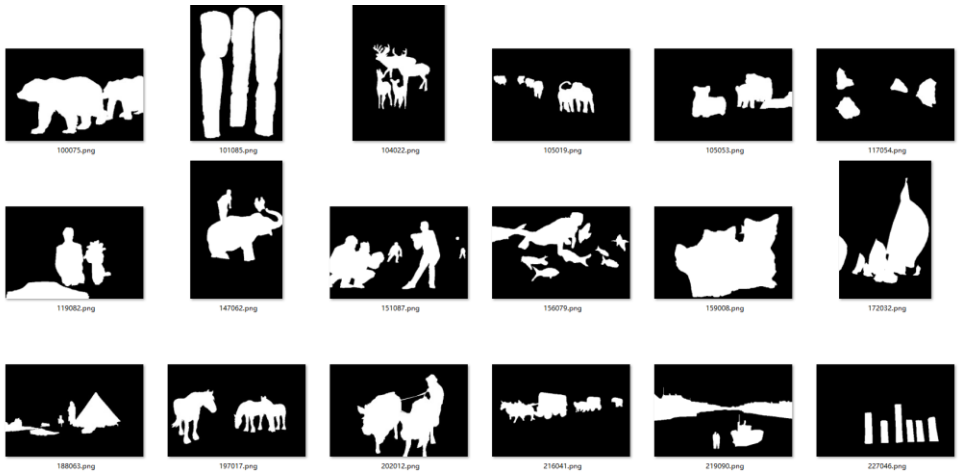


Figure 3-3 Groundtruth examples in our proposed MSOD dataset.

As depicted in Figure 3-2 and Figure 3-3, we provide examples of the images and their corresponding ground truths. It is clear that all the images presented encompass multiple salient objects. The multiple salient objects in each image raise the difficulty level, which pose a challenging scenario for all the SOD models. Therefore, our proposed dataset can be used to test the robustness and effectiveness of current state-of-the-art models and our proposed method in handling complex visual scenarios.

3.5 Proposed Method

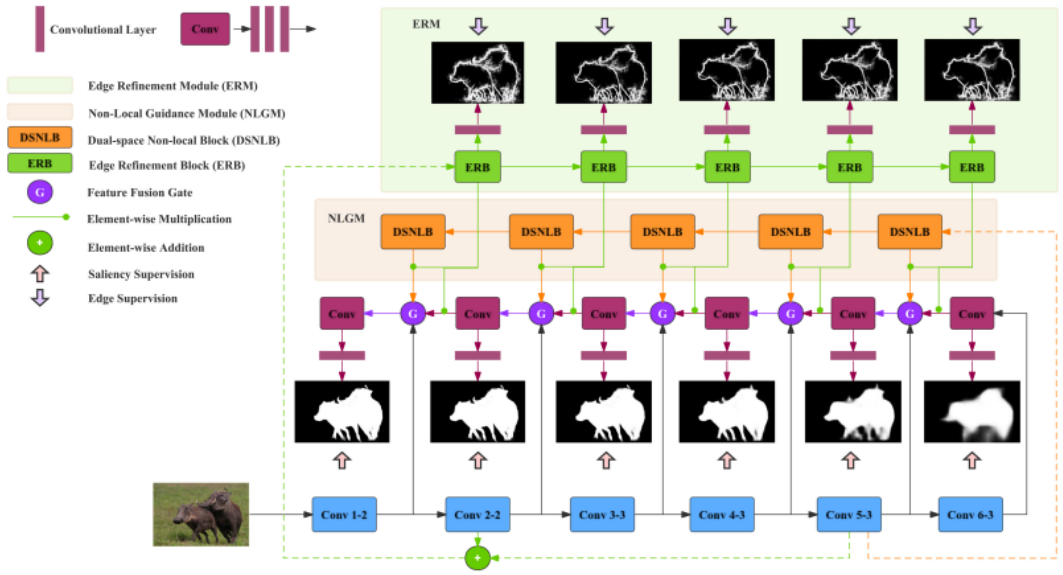


Figure 3-4 The overall pipeline of our proposed approach, here shown using a VGG backbone. The red, orange, and green boxes capture saliency features, non-local features, and edge features respectively. Element-wise multiplication operates between each pair of ERB-DSNLB (edge and non-local features) and ERB-Conv (edge and saliency features). Our final prediction map is generated based on the fusion of 6 multi-scale saliency features in top-down pathway.

The architecture of our proposed method is shown in Figure 3-4. Our model is based on a U-shaped Fully Convolutional Network (FCN) that incorporates a bottom-up pathway (the backbone) and a top-down pathway. Similar to most deep SOD models, we employ the VGG network to demonstrate our proposed structure. Following the approach of EGNet [21] and DSS [1], we truncate the last three fully connected layers and connect an additional side path to the last pooling layer of VGG. This provides six outputs from the bottom-up pathway, representing the multi-level features captured from Conv1-2 to Conv6-3, which can be defined as a feature set $S = \{S^1, S^2, S^3, S^4, S^5, S^6\}$.

The top-down pathway processes multi-scale saliency features through a series of convolutional blocks, each consisting of three convolutional layers and *ReLU* activations. This saliency feature set can be represented as

$F = \{F^1, F^2, F^3, F^4, F^5, F^6\}$, where F^6 is the saliency feature output by the sixth convolutional block (the rightmost Conv in Figure 3-4), and so forth.

Additionally, we apply deep supervision to each enhanced feature F^i using a cross-entropy loss. A convolutional layer D_F^i is used on each enhanced feature to transform multi-channel features into a single-channel prediction map. Therefore, the supervision can be expressed as:

$$L_F^i(F^i; W_{DF}^i) = -\sum_{j \in Y^+} logPred(y_j = 1 | F^i; W_{DF}^i) - \sum_{j \in Y^-} logPred(y_j = 0 | F^i; W_{DF}^i), i \in [1,6] \quad (3-9)$$

where $Pred(y_j = 1 | F^i; W_{DF}^i)$ denotes the prediction map and each value demonstrates the salient region confidence for the pixel. Y^+ and Y^- denotes the salient pixels set and the non-salient pixels set respectively, while W_{DF}^i denotes the parameters of the convolutional layers D_F^i.

3.5.1 Non-Local Guidance Module

In this module, we model long-range dependencies in both the spatial and channel spaces. Inspired by Fu et al., 2019 [37], we incorporate dual-space non-local information within two parallel pathways that capture pixel-wise contextual information and channel-wise relationships. Unlike Fu et al., 2019 [37], who directly appended a single attention module on top of a Fully Convolutional Network (FCN) for scene segmentation, our NLGM comprises 5 stacked Dual-Space Non-Local Blocks (DSNLBs), each at a different stage of the top-down pathway. We select the feature map S^5 extracted from Conv5-3 as the input for the NLGM, as it holds high-level semantic information and still contains more spatial information than S^6. Figure 3-5 illustrates the detailed structure of the first DSNLB.

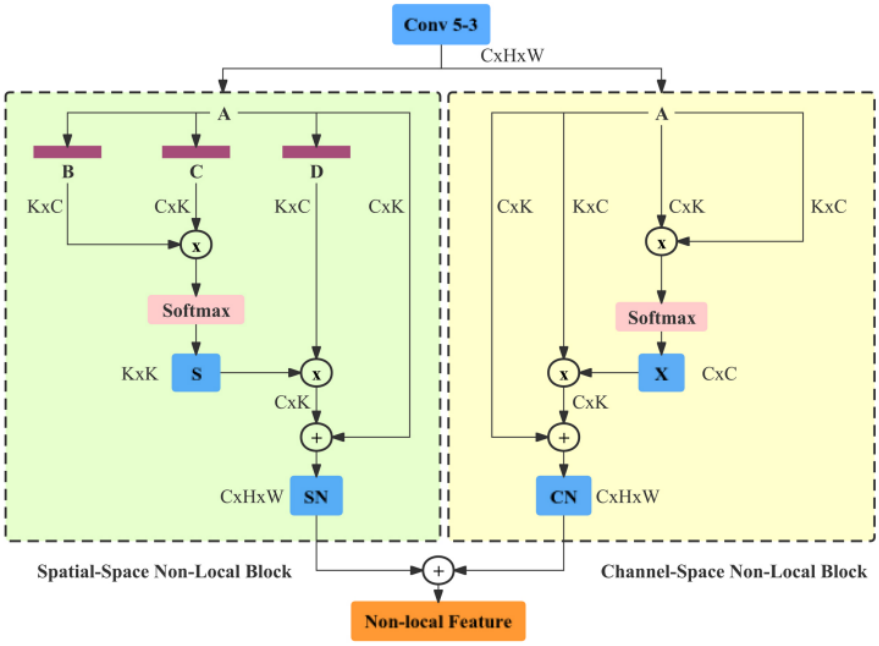


Figure 3-5 The architecture of a dual-space non-local block (DSNLB). C, H and W demonstrate the channel number, height and width of given feature map respectively and K = H × W.

3.5.1.1 Spatial-Space Non-Local Block

For a given feature map $A \in R^{C\times H\times W}$, A will be firstly fed into 3 convolutional layers to generate three feature maps $\{B, C, D\} \in \mathrm{R}^{\mathrm{CxHxW}}$. Then, B, C and D will be reshaped to $R^{C\times K}$ and K is the number of pixels in a channel, e.g., $K = H \times W$. After that, a matrix multiplication operation between the transpose of B and C will be done and followed by a *softmax* layer to get the spatial attention map $S \in R^{(HxW)x(HxW)}$, which can be defined as follows:

$$s_{ji} = \frac{exp(B_i \cdot C_j)}{\sum_{i=1}^{K} exp(B_i \cdot C_j)} \qquad (3-10)$$

where s_{ji} here indicates the i^{th} position's response on j^{th} position. In this case, if the similarity between two positions is high, a greater correlation will be witnessed.

Meanwhile, a matrix multiplication will be done between the transpose of S and D, of which the result will be reshaped to R^{CxHxW}. Different from the work in [37], we only use the residual operation without the weight parameter to get $SN \in R^{CxHxW}$:

$$SN_j = \sum_{i=1}^{K} (s_{ji} D_i) + A_j \qquad (3-11)$$

It can be found that the value of each position of SN is a weighted sum of all the positions.

3.5.1.2 Channel-Space Non-Local Block

A is firstly reshaped to $R^{C\times K}$. Then, a matrix multiplication between A and the transpose of it has been done. After a *softmax* layer, the channel attention map will be obtained, which can be defined as:

$$x_{ji} = \frac{exp(A_i \cdot A_j)}{\sum_{i=1}^{C} exp(A_i \cdot A_j)} \qquad (3-12)$$

where x_{ji} demonstrates the i^{th} channel's response on j^{th} channel. Then, a matrix multiplication between the transpose of X and A are done followed by a reshape operation. Similar to position attention module, an element-wise sum operation is used to obtain the final output $CN \in R^{CxHxW}$:

$$CN_j = \sum_{i=1}^{C} (x_{ji} A_i) + A_j \qquad (3-13)$$

Finally, an element-wise sum operation is carried out between the outputs from Spatial Attention Non-Local Block and Channel Attention Non-Local Block and followed by a convolutional layer D_N^i to get the final result N^i:

$$N^i = D_N^i (SN^i + CN^i) \qquad (3-14)$$

Multi-Hop Communications:

The NLGM in our model comprises five Dual-Space Non-Local Blocks (DSNLBs). Each of these blocks generates non-local features, denoted as

N^i, at each stage of the top-down pathway. By stacking several DSNLBs, these non-local features are progressively refined through a multi-hop communication mechanism between features that share affinity in both channel and spatial dimensions.

In this manner, saliency-specific semantic information is distributed across the image space and feature space. The non-local features, with a global view, are better equipped to handle the complexities of diverse scenes and the detection of multiple salient objects. This is facilitated by their enhanced receptive field, allowing them to process a larger context within the image and thus improve the robustness and performance of SOD tasks, especially in complex scenarios with multiple salient objects.

3.5.2 Feature Fusion

3.5.2.1 Edge Refinement Module

Inspired by the commonly used boundary detection technique in SOD models (Qin et al., 2019 [19]; Zhao et al., 2019 [21]), salient edge features are incorporated into the feature fusion gate to support the training of non-local and salient features.

It is acknowledged that the low-level features in shallower layers typically possess spatial information, including attributes like edges and corners, useful for recreating object boundaries. In comparison, the complex, high-level features in the deeper layers hold semantic information that's ideal for identifying the salient object. Consequently, we combine the low-level feature S^2 and high-level feature S^5 to serve as input for the Edge Refinement Guiding Module, which can be symbolized as:

$$E_{input} = S^2 + Up(S^5; S^2) \qquad (3-15)$$

where $Up\ (*;\ S^2)$ is the bilinear interpolation operation used to up-

sample $*$ to have the same size as S^2.

Each Edge Refinement Block consists of a convolutional layer followed by a ReLU layer to enhance the edge feature. For simplicity, we denote the corresponding convolutional layer and ReLU layer with C_E^i. In the Edge Refinement Guiding Module, we stack five Edge Refinement Blocks and the resulting edge feature from each block can be denoted as:

$$E^i = \begin{cases} C_E^i(E^{i+1}), i = 1,2,3,4 \\ C_E^5(E_{input}), i = 5 \end{cases} \quad (3-16)$$

Meanwhile, to effectively capture the edge information, we employ intermediate supervision for the edge feature. A convolutional layer, denoted as D_E^i, is utilized to transform the generated edge feature into a single-channel prediction map. The supervision here can be expressed as:

$$L_E^i(E^i; W_{DE}^i) = -\sum_{j \in Z^+} logPred(y_j = 1 | E^i; W_{DE}^i)$$

$$-\sum_{j \in Z^-} logPred(y_j = 0 | E^i; W_{DE}^i), i \in [1,5] \quad (3-17)$$

where Z^+ and Z^- denote the edge pixels set of salient regions and background pixels set respectively. W_{DE}^i denotes the parameters of the convolutional layer D_E^i.

3.5.2.2 Feature Fusion Gate

The majority of current SOD models directly fuse various features without distinction, which can introduce redundancy and, to some extent, undermine the performance of SOD models. Hence, it is vital to filter out redundant information and emphasize the useful information. As demonstrated in Figure 3-6, our proposed model incorporates a Feature Fusion Gate (FFG) to selectively fuse features drawn from three distinct sources.

Salient edge features are integrated with salient and non-local features through element-wise multiplication. This procedure aims to emphasize the

activations that are shared between feature maps, promoting complementarity between the non-local, saliency and edge features. Features that align across modules will experience accelerated training, with activations of those that are less relevant to other blocks reduced. The subsequent salient and non-local features are then merged using channel-wise attention. Firstly, we standardize the spatial size and channel count:

$$N_{Refined}^{i} = Up\left(Trans\left(N^{i}; S^{i}\right); S^{i}\right), i\epsilon[1,5] \quad (3-18)$$

$$F_{Refined}^{i} = Up\left(Trans\left(F^{i}; S^{i}\right); S^{i+1}\right), i\epsilon[2,6] \quad (3-19)$$

$$E_{Guiding}^{i} = Up\left(Trans\left(E^{i}; S^{i}\right); S^{i}\right), i\epsilon[1,5] \quad (3-20)$$

where $Up\left(*; S^{i}\right)$, $Trans\ (*;\ S^{i})$ denote Up-sampling the feature map $*$ to has the same size as S^{i} and convert the channel number of $*$ to has the same channel number of S^{i} respectively.

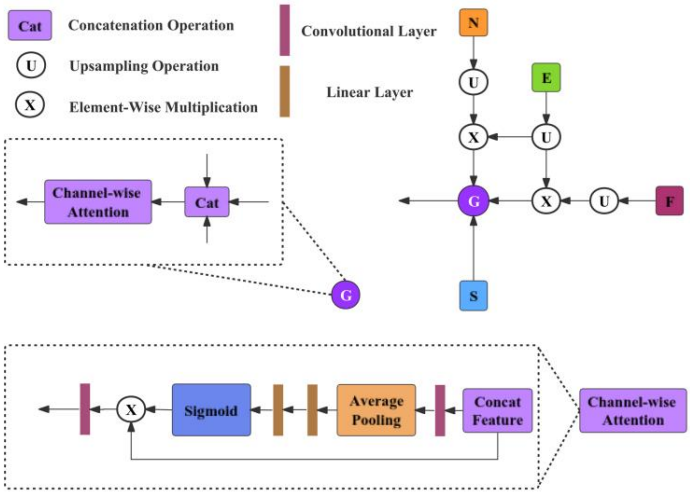


Figure 3-6 The structure of a feature fusion gate. N, E, F and S demonstrate non-local feature, edge feature, saliency feature and the corresponding side output of bottom-up pathway respectively.

Then, the fused feature F_{fusion}^{i} can be defined as:

$$F_{fusion}^{i} = CA\left(Cat\left(N_{Refined}^{i} \otimes E_{Guiding}^{i}, F_{Refined}^{i+1} \otimes E_{Guiding}^{i}, S^{i}\right)\right), i\epsilon[1,5] (3-21)$$

where $\otimes$ denotes the element-wise multiplication, Cat denotes the concatenation operation and CA denotes the channel-wise attention

operation, which can be formulated as:

$$CA(*, \theta_{ca}) = * \cdot \left(\sigma \left(fc_2 \left(\delta \left(fc_1 \left(ap(*, \theta_1) \right) \right), \theta_2 \right) \right) \right) \qquad (3-22)$$

Here, θ_{ca} demonstrates the parameters in channel-wise attention, while ap stands for a global average pooling layer and fc refers to a fully-connected layer. σ and δ represent the sigmoid function and ReLU functions respectively. Here, the feature fusion gate offers a mechanism to choose the most beneficial channels for saliency from each module, thereby fusing features in a distinctive manner.

As shown in Figure 3-7, our non-local features can highlight the salient object locations across the visual scene, while the edge features can clearly define the boundaries of different objects. By selectively fusing different features, multiple salient objects are correctly defined with clear boundaries.

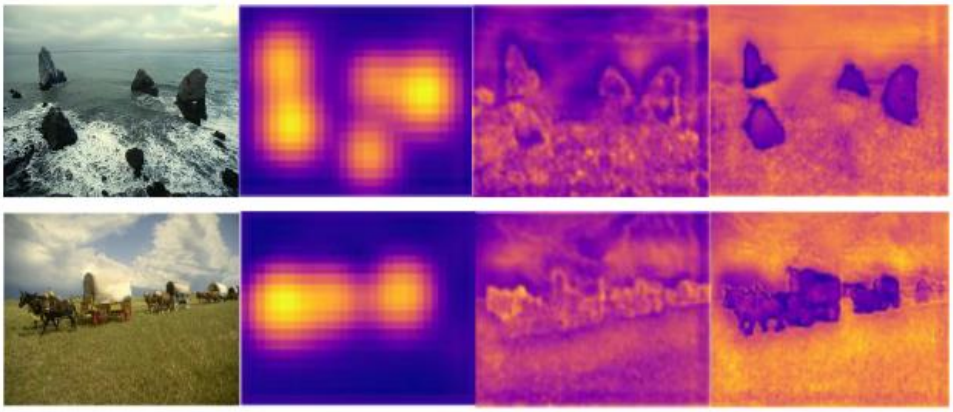


Figure 3-7 Feature visualization of non-local features, edge features and the refined features after feature fusion.

3.5.3 Saliency Inference

To maximize the use of multi-scale saliency features, we generate the final prediction map in a hierarchical fashion based on the fusion of six saliency features F^i, in a coarse-to-fine manner. This multi-scale fusion strategy also helps reduce the risk of missing salient objects within multi-saliency visual

scenes. Complementary features F^2, F^3, F^4, F^5, F^6 are upsampled and convolved to match the spatial and feature size of F^1. They are then combined using element-wise addition to generate a final feature F_{final}. A convolutional layer, D_{final}, is used to transform the feature map F_{final} into a single-channel prediction map, which is trained using cross entropy:

$$L_{final}(F_{final}; W_{Dfinal}) = -\sum_{j \in Y^+} logPred(y_j = 1 | F_{final}; W_{Dfinal}) - \sum_{j \in Y^-} logPred(y_j = 0 | F_{final}; W_{Dfinal}) \quad (3-23)$$

where Y^+ and Y^- denotes the salient pixels set and the non-salient pixels set respectively and W_{Dfinal} denotes the parameters of the convolutional layers D_{final}.

Therefore, the total loss of the proposed model can be expressed as:

$$L_{Total} = \sum_{i=1}^{i=6} L_F^i(F^i; W_{DF}^i) + \sum_{i=1}^{i=5} L_E^i(E^i; W_{DE}^i) + L_{final}(F_{final}; W_{Dfinal}) \quad (3-24)$$

3.6 Experiment

3.6.1 Datasets and Evaluation Metrics

In order to demonstrate the performance of our proposed approach, we evaluate our model using five commonly used benchmark datasets. These include DUT-OMRON (Yang et al., 2013 [7]), HKU-IS (Li and Yu, 2015 [61]), DUTS (Wang et al., 2017 [59]), ECSSD (Yan et al., 2013 [60]), SOD (Movahedi and Elder, 2010 [62]) and our proposed dataset MSOD.

For evaluating performance, we employ three broadly used evaluation metrics: F-measure, mean absolute error (MAE), and the structure-based metric S-measure [63]. Here, the F-measure is a weighted combination of precision and recall, which can be articulated as:

$$F_\beta = \frac{(1+\beta^2) \times Precision \times Recall}{\beta^2 \times Precision + Recall} \qquad (3-25)$$

Following most of the SOD methods [21] [67] [23] [68] [25], β^2 is assigned a value of 0.3 to place a higher emphasis on precision. Following most of the SOD models, we report the maximum F_β derived from all precision-recall pairs. Nonetheless, F-measure doesn't evaluate the true negative pixels. To address this issue, we use MAE to calculate the mean absolute error on a pixel level, which can be defined as:

$$MAE = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} |P(i,j) - Y(i,j)| \qquad (3-26)$$

Here, W and H represent the width and the height of the images, while P and Y stand for the prediction map and the ground truth, respectively.

The above-mentioned metrics are computed at the pixel-wise level, which may not completely capture the structural information. The S-measure is designed to assess the region-aware S_r and object-aware S_o structural similarities between the real-valued saliency map and the binary ground truth, which can be articulated as:

$$S = \alpha \times S_0 + (1-\alpha) \times S_r \qquad (3-27)$$

Where α is empirically set to 0.5.

3.6.2 Implementation Details

Our proposed method is implemented in PyTorch and trained on the DUTS-TR dataset. The salient edge ground truth is calculated using the Sobel operator. To compare our method against other state-of-the-art techniques, we train our model using both VGG and ResNet-50 as backbones. The parameters of these backbones are initialized using pretrained models on ImageNet [64], while the weights of the newly added layers are randomly

initialized. We utilize the Adam optimize [65] with an initial learning rate of 2e-5, which is reduced by a factor of 10 after 30 epochs. Our model is trained for a total of 40 epochs, a process which typically takes three days on a single 2080Ti GPU, with a forward pass taking approximately 0.02 seconds.

3.6.3 Quantitative Comparisons with the State-of-the-Art

We compare our proposed method against 14 recent state-of-the-art methods: DSS (Hou et al., 2017 [1]), BDMP (Zhang et al., 2018a [17]), PAGR (Zhang et al., 2018b [18]), RAS (Chen et al., 2018 [26]), BASNet (Qin et al., 2019 [19]), AFNet (Feng et al., 2019 [20]), PiCANet (Liu et al., 2018 [66]), PoolNet (Liu et al., 2019 [22]), EGNet (Zhao et al., 2019 [21]), CPD (Wu et al., 2019a [67]), SCRN (Wu et al., 2019b [23]), GateNet (Zhao et al., 2020 [68]), MINet (Pang et al., 2020 [25]), and SCWSSOD (Yu et al., 2021 [69]). To ensure a fair comparison, all the saliency maps of competing methods were either generated by pre-trained models or pre-produced by the respective authors. Note the competitive SOD methods are also trained on DUTS-TR dataset.

Model	ECSSD 1000 images			DUTS-TE 5019 images			HKU-IS 1447 images		
	MaxF ↑	MAE ↓	S ↑	MaxF ↑	MAE ↓	S ↑	MaxF ↑	MAE ↓	S ↑
VGG-Backbone									
DSS (CVPR2017)	0.9207	0.0517	0.8821	0.8251	0.0565	0.8237	0.9161	0.0401	0.8783
BDMP (CVPR2018)	0.9284	0.0446	0.9109	0.8514	0.049	0.8616	0.9205	0.0389	0.9065
PAGR (CVPR2018)	0.9259	0.0608	0.8883	0.854	0.0555	0.8383	0.9187	0.0475	0.8891
RAS (ECCV2018)	0.9211	0.0564	0.8928	0.8311	0.0594	0.8385	0.9128	0.0454	0.8874
BASNet (CVPR2019)	0.9425	0.037	0.9162	0.8594	0.0476	0.8656	0.9297	0.0329	0.9077
AFNet (CVPR2019)	0.935	0.0418	0.9134	0.8628	0.0458	0.8666	0.9252	0.0355	0.9058
Ours	0.9485	**0.0344**	0.9261	**0.8894**	0.0381	**0.8878**	0.935	**0.03**	0.9167
ResNet-Backbone									
PiCANet (CVPR2018)	0.9349	0.0464	0.917	0.8597	0.0506	0.8686	0.9193	0.0437	0.9045
PoolNet (CVPR2019)	**0.9489**	0.035	**0.9263**	**0.8891**	**0.0368**	0.8865	**0.9358**	**0.03**	**0.9187**
EGNet (ICCV2019)	0.9474	0.0374	0.9247	0.8885	0.0392	**0.8868**	0.9352	0.0309	0.9179
CPD (CVPR2019)	0.9393	0.0371	0.9181	0.8653	0.0434	0.8689	0.9252	0.0339	0.9064
SCRN (ICCV2019)	**0.9496**	0.0375	**0.9272**	0.8875	0.0398	0.8847	0.9351	0.0332	0.9169
GateNet (ECCV2020)	0.9454	0.0401	0.9198	0.8873	0.0401	0.8847	0.9334	0.0331	0.9153
MINet (CVPR2020)	0.9475	**0.0335**	0.9249	0.8836	**0.0372**	0.8837	**0.9353**	**0.0283**	**0.9197**
SCWSSOD* (AAAI2021)	0.9145	0.0489	0.8818	0.844	0.0487	0.8405	0.9111	0.0375	0.8824
Ours	**0.9519**	**0.0325**	**0.9297**	**0.8967**	**0.0358**	**0.8946**	**0.9389**	**0.0293**	**0.9218**

Table 3-1 Quantitative comparison with other state-of-the-art methods on 3 widely used relatively easy datasets. ↑ and ↓ indicate higher or lower is better respectively and * denotes weakly-supervised methods. The best three results among both backbones are marked as red, blue and cyan. Our method achieves top results under 3 evaluation metrics across all datasets without any pre-processing and post-processing.

As shown in Table 3-1, it demonstrates the quantitative comparison with other state-of-the-art methods based on 3 relatively easy dataset: ECSSD, DUTS-TE, HKU-IS.

3.6.3.1 ECSSD

In terms of MaxF, our proposed Resnet-based method rank 1st , while the SCRN method rank 2nd, following by the PoolNet, with the MaxF scores being 0.9516, 0.9496, 0.9489 respectively. When it comes to MAE, our proposed ResNet-based method and VGG-based method rank 1st and 3rd

(0.0325 and 0.0344 respectively), MINet ranks the 2nd (0.0355). Regarding S-measure, the top-performing models are based on the ResNet-Backbone architectures. Our model secured the first place with an impressive S-measure of 0.9297. Not far behind was the SCRN mode, achieving an S-measure of 0.9272. The third position was held by the PoolNet mode, with an S-measure of 0.9263.

3.6.3.2 DUTS-TE

For the MaxF measure, the model with the highest performance is ours using the ResNet-Backbone architecture, achieving a MaxF score of 0.8967. This is closely followed by our VGG-based model, which gets a MaxF measure of 0.8894. The third highest MaxF performance is held by the PoolNet model, scoring 0.8891. Looking at the MAE measure, the model with the best performance is again ours with the ResNet-Backbone architecture, having an MAE score of 0.0358. PoolNet is the next best performing model, with an MAE score of 0.0368. MINet takes the third spot with an MAE measure of 0.0372. As for the S measure, the top performing model is ours with the ResNet-Backbone architecture, yielding an S score of 0.8946. Followed by our VGG-based model with an S measure of 0.8878. The third spot is taken by the EGNet model, with an S measure of 0. 8868.

3.6.3.3 HKU-IS

For the MaxF measure, the top performing model is ours with the ResNet-Backbone architecture, achieving a MaxF score of 0.9389. Following in second place is PoolNet, with a MaxF score of 0.9358. In third place is MINet, scoring 0.9353 in MaxF. Regarding the MAE measure, where lower is better, MINet with the ResNet-Backbone architecture leads the group with an MAE score of 0.0283. The second best performance in this category is seen from our model, with an MAE measure of 0.0293. The third place is

held by PoolNet and our VGG model together with same MAE score of 0.0300. In terms of the S measure, our model with the ResNet-Backbone architecture once again comes out on top with an S score of 0.9218. Following closely in second place is MINet, scoring 0.9197 in S measure. PoolNet takes the third position with an S score of 0.9187.

Model	DUT-O 5168 images			SOD 300 images			MSOD 300 images		
	MaxF ↑	MAE ↓	S ↑	MaxF ↑	MAE ↓	S ↑	MaxF ↑	MAE ↓	S ↑
VGG-Backbone									
DSS (CVPR2017)	0.7812	0.0628	0.7899	0.841	0.1201	0.7478	0.824	0.055	0.7806
BDMP (CVPR2018)	0.7739	0.0636	0.8091	0.8517	0.1057	0.7833	0.8401	0.0538	0.8379
PAGR (CVPR2018)	0.7706	0.0709	0.7751	0.8358	0.1447	0.7137	0.8204	0.0627	0.7852
RAS (ECCV2018)	0.7864	0.0617	0.8141	0.8473	0.1225	0.7608	0.837	0.0597	0.8167
BASNet (CVPR2019)	0.8052	0.0565	0.8361	0.8487	0.1119	0.766	0.8396	0.0541	0.8306
AFNet (CVPR2019)	0.797	0.0573	0.8258	0.8499	0.1087	0.77	0.8276	0.0547	0.8191
Ours	**0.8194**	0.0541	**0.8421**	**0.8761**	0.0996	0.7922	0.8531	0.0478	0.841
ResNet-Backbone									
PiCANet (CVPR2018)	0.8027	0.0653	0.8318	0.8528	0.1024	0.7871	0.819	0.0641	0.8223
PoolNet (CVPR2019)	0.8048	**0.0539**	0.8309	0.8706	0.1034	0.7854	**0.8546**	**0.0459**	**0.8429**
EGNet (ICCV2019)	0.8152	**0.0531**	**0.8408**	**0.8778**	**0.0969**	**0.8**	0.8516	**0.047**	0.8402
CPD (CVPR2019)	0.7964	0.056	0.8247	0.8568	0.1095	0.7646	0.8241	0.0539	0.8109
SCRN (ICCV2019)	0.8112	0.056	0.8364	0.8655	0.1046	0.7851	0.8384	0.0527	0.8244
GateNet (ECCV2020)	**0.8178**	0.0549	0.838	0.8731	0.0981	0.7948	**0.8623**	0.0483	**0.8507**
MINet (CVPR2020)	0.8097	0.0555	0.8325	0.873	**0.0905**	**0.7973**	0.8472	0.0474	0.84
SCWSSOD* (AAAI2021)	0.7823	0.0602	0.8117	0.8367	0.1077	0.7503	0.8329	0.0534	0.806
Ours	**0.8234**	**0.053**	**0.847**	**0.8786**	**0.0934**	**0.8024**	**0.872**	**0.0442**	**0.8614**

Table 3-2 Quantitative comparison with other state-of-the-art methods on 2 widely used relatively hard datasets and the proposed MSOD dataset. ↑ and ↓ indicate higher or lower is better respectively and * denotes weakly-supervised methods. The best three results among both backbones are marked as red, blue and cyan. Our method achieves top results under 3 evaluation metrics across all datasets without any pre-processing and post-processing.

Table 3-2 demonstrates the quantitative comparison with other state-of-the-art methods based on 3 relatively difficult datasets, most of which include challenging scenes, viz., multiple salient objects, connected salient

objects and so on.

3.6.3.4 DUT-O

For the MaxF measure, the highest performing model is our model with the ResNet-Backbone architecture, achieving a MaxF score of 0.8234. The second highest MaxF performance is achieved by our VGG-based model, with a score of 0.8194. The GateNet model holds the third position, scoring 0.8178 in MaxF. In terms of the MAE measure, our model with the ResNet-Backbone architecture again shows the best performance with an MAE score of 0.053. Following closely is the EGNet model, achieving an MAE measure of 0.0531. The third-best model in MAE is PoolNet, with an MAE score of 0.0539. Regarding the S measure, our model with the ResNet-Backbone architecture remains the top performer with an S score of 0.847. Coming in second is our model with the VGG-Backbone architecture, showing an S measure of 0.8421. The third position is held by EGNet, with an S score of 0.8408.

3.6.3.5 SOD

Regarding the MaxF measure, the top performer is our model using the ResNet-Backbone architecture, achieving a MaxF score of 0.8786. The second-highest performer is EGNet, with a MaxF score of 0.8778. The third spot is taken by our VGG model, achieving a MaxF score of 0.8761. In terms of the MAE measure, MINet using the ResNet-Backbone architecture leads with an MAE score of 0.0954. The second-best performance is by our model, with an MAE score of 0.0934. The third place is occupied by EGNet, with an MAE score of 0.0969. Looking at the S measure, our model using the ResNet-Backbone architecture again outperforms others, scoring 0.8024. The second position is held by EGNet, with an S score of 0.8. The third spot is occupied by MINet, presenting an S score of 0.7973.

3.6.3.6 MSOD

With respect to the MaxF measure, the best performing model is ours, achieving a MaxF score of 0.872. This is followed by GateNet, which scores 0.8623 in the MaxF measure. The third place is taken by PoolNet with a MaxF score of 0.8546. In terms of the MAE measure, our model using the ResNet-Backbone architecture again comes out on top, achieving an MAE score of 0.0442. Following that is PoolNet with an MAE score of 0.0459. The third spot is occupied by EGNet, which scores 0.047 in the MAE measure. Finally, looking at the S measure, our model using the ResNet-Backbone architecture is once again the top performer with an S score of 0.8614. GateNet takes the second spot with an S score of 0.8507, and the third spot is occupied by PoolNet, scoring 0.8429 in the S measure.

In summary, compared to the current leading models: MINet, PoolNet, EGNe, and SCRN, our proposed approach shows an average enhancement of 0.92%, 2.27%, 2.43%, and 4.09% respectively across five commonly used datasets. When we turn our attention to the MSOD dataset, the average advancement over these methods rises to 4.08%, 2.64%, 3.63%, and 8.21% respectively. These results demonstrate the robust capabilities of our proposed method in MSOD, achieving state-of-the-art performance on this challenging dataset.

3.6.4 Precision-Recall Curves Comparison

In Figure 3-8, we present the comparison showing the precision-recall curves derived from three widely recognized SOD datasets, as well as from our newly proposed MSOD dataset.

When analyzing precision-recall curves, the orientation towards the top-right corner of the graph plays a significant role in determining the model's performance. The closer a model's values fall to this top-right corner,

the better it is considered to be. The reasoning behind this is twofold. Firstly, precision, represented on the y-axis, refers to the fraction of relevant instances among the retrieved instances. A higher value of precision indicates that the model has a lower rate of false positives, which means that it is more accurate in its predictions. Secondly, recall, plotted on the x-axis, is a measure of the model's ability to find all the relevant cases within a dataset. A higher recall value suggests that the model is doing well in identifying true positives and minimizing the risk of false negatives. Thus, a model with values closer to the top-right side of the precision-recall curve excels in both precision and recall, indicating that it maintains a strong balance between these two metrics, where the model is not only capable of accurately identifying true positives but also ensures that it identifies most of the relevant cases, resulting in fewer missed detections or false negatives. Hence, in terms of model evaluation and comparison, those with precision-recall curve values leaning towards the top-right corner are considered superior in performance.

Our proposed method stands out from the rest due to its exceptional performance across the majority of the thresholds. This implies that our approach not only effectively identifies the salient objects in the majority of the cases, but it also maintains a low false-positive rate, leading to a higher precision score.

The superiority of our proposed method becomes particularly clear when applied to the two largest datasets - DUT-OMRON and DUTS-TE. These two datasets are commonly used because of their size, and the fact that our method outperforms others on these two demonstrates its scalability and robustness. This suggests that our method is adaptable and reliable.

When the analysis turns to the MSOD dataset, which is challenging due to the multiple salient objects, the gap in performance becomes even more broadens. Our proposed method begins to widen the margin and sets a new benchmark, significantly outperforming other techniques. This enhanced

performance on the MSOD dataset illustrates the method's capability to handle complex and challenging scenarios that involve multiple salient objects. The superior results obtained from this dataset demonstrate that our proposed technique is well-equipped to deal with intricate visual scenarios.

Therefore, it is evident that our approach not only surpasses others on standard datasets but truly excels when the complexity and difficulty of the task increase. This adaptability to task difficulty makes our method a reliable and effective solution for multiple salient objects. The strong performance on the MSOD dataset suggests promising potential for real-world applications, where the scenarios could be as complex or even more challenging than those presented in this dataset.

This all-around performance across various thresholds and datasets reaffirms our confidence in the robustness, reliability, and precision of our proposed method in SOD.

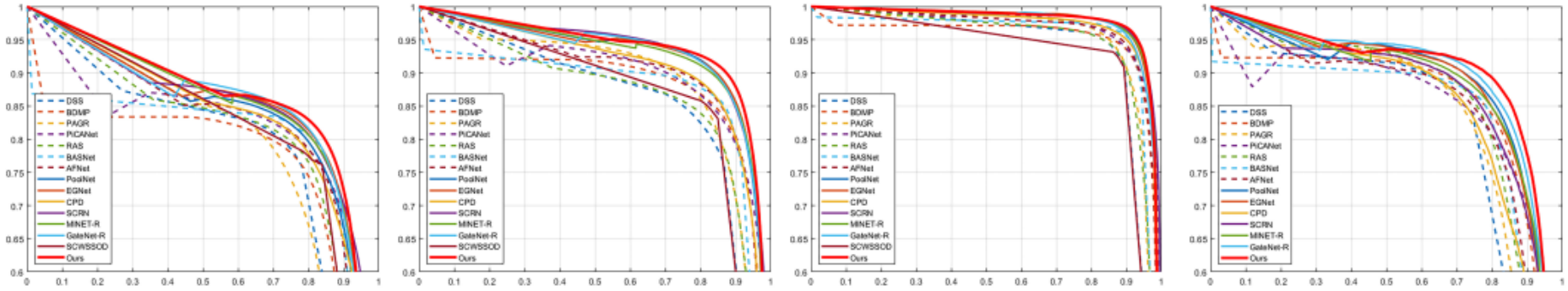


Figure 3-8 Precision (vertical axis) recall (horizontal axis) curves on three popular salient object detection datasets and the proposed MSOD dataset. The red solid line demonstrates our proposed method.

3.6.5 Visual Comparison

The effectiveness of our approach can be visually assessed in Figure 3-9. It shows that our method provides excellent performance in the images containing multiple salient objects. Utilizing non-local features along with a top-down feature fusion strategy enables our system to thoroughly exploit the long-range dependencies between salient objects. This strategy allows for the sharing information from multiple salient objects of the image, leading to more comprehensive image processing.

To be more precise, the first row of Figure 3-9 demonstrates a challenging scenario with very low contrast. MINet, SCRN, RAS and BASNet all get missed salient objects because of the low contrast. In comparison, although other methods can detect both salient objects, the quality of these saliency maps are very low, e.g., missing parts of objects or unclear edges. Compared to other methods, our proposed method generates very high-quality saliency maps with clear boundaries, this suggests our use of non-local features can help the proposed method accurately locate the salient objects and the edge guidance module embedded in the feature fusion gate works well for refining the objects' boundaries.

Regarding the second row of Figure 3-9, all the other models miss salient objects. Our proposed method not only detects all the salient objects, but also has clear detail information, which indicates the importance of utilizing the non-local features and edge features in our proposed method.

The third row and the fifth row of Figure 3-9 also show challenging visual scenes, where small salient objects separated in the visual scene with low contrast. The other state-of-the methods either always get missing salient objects or even cannot make any predictions in these images. In comparison, our proposed method generates very good saliency maps including all the small salient objects. This also demonstrates the strong performance of our proposed method in dealing with the MSOD problem

even in complicated visual scene.

The fourth row of Figure 3-9 introduces a scenario where several small salient object clustered. We can see from the results that other methods can always bring some redundant information between different small salient objects. This demonstrates the effectiveness of our proposed feature fusion gate, which can selectively choose the useful information between different features, therefore generating high-quality saliency maps without redundant information.

In summary, by making full use of long-range dependencies and edge features based on features fusion gate, our proposed method can effectively highlight salient objects across the entire image with clear boundaries. This facilitates a more comprehensive view of the image, as information can be effectively relayed and interchanged between separate regions of the image with the redundant information filtered out.

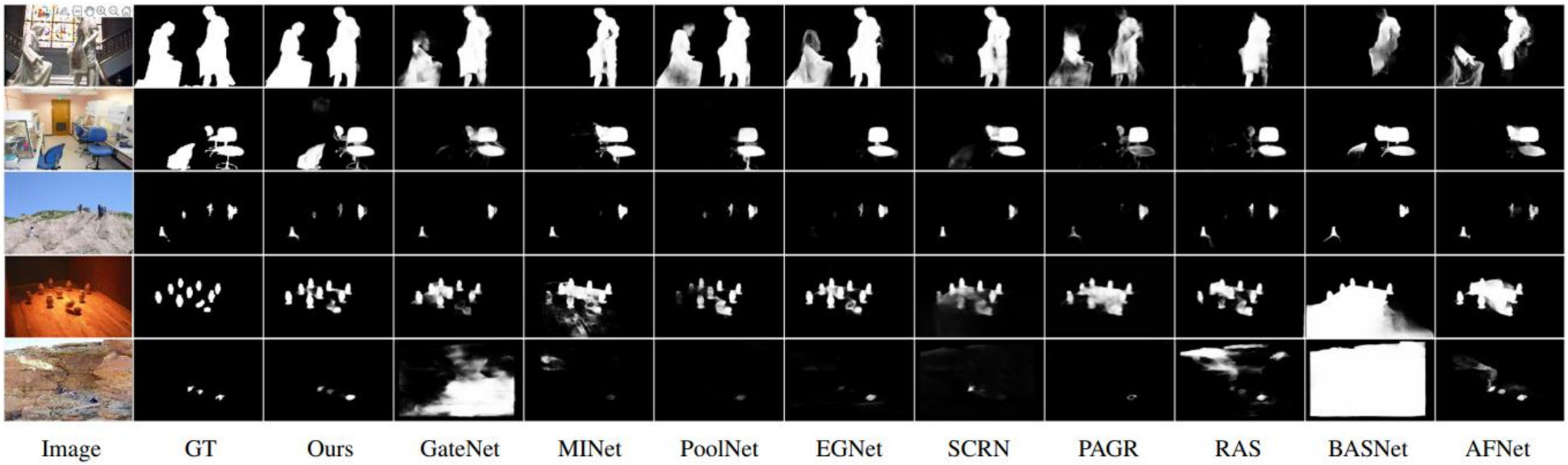


Figure 3-9 Qualitative comparisons with state-of-the-art approaches over some of the challenging images. The main object classes are statuette, chairs, human and bowling.

3.6.6 Ablation Studies

In the following section, we make exploration of the specific contributions made by various components of our proposed model. All the experiments we conduct on the two largest datasets, namely DUTS-TE and DUT-OMRON. Following PoolNet (Liu et al., 2019 [22]), EGNet (Zhao et al., 2019 [21]), we conduct the experiments based on VGG.

Models				DUTS-TE			DUT-OMRON		
NLGM	FFG	ERM	E	MaxF ↑	MAE ↓	S ↑	MaxF ↑	MAE ↓	S ↑
				0.8761	0.0425	0.875	0.7958	0.0575	0.8276
✓				0.8836	0.041	0.8809	0.8103	0.0562	0.8357
✓		✓		0.8847	0.041	0.8839	0.8138	0.0563	0.8366
✓	✓			0.8858	0.0396	0.8857	0.8153	0.056	0.8379
✓	✓		✓	0.8882	0.0395	0.8862	0.8175	0.0543	0.8407
✓	✓	✓		**0.8894**	**0.0381**	**0.8878**	**0.8194**	**0.0541**	**0.8421**

Table 3-3 Ablation analysis of different components in our proposed architecture.

3.6.6.1 Effectiveness of NLGM

When set against a fundamental baseline structure (as shown in the first row of Table 3-3), the utilization of non-local guidance module (indicated in the second row) results in enhancements in performance across all evaluative metrics for both datasets. These gains indicate the effectiveness of our NLGM in successfully capturing long-range dependencies. Precisely, the comparison provides clear evidence that the addition of non-local guidance can improve the accuracy and efficiency of the system. As opposed to the basic structure, the integration of non-local features allows the proposed model to detect and analyze relationships between objects in an image that

aren't stay close. This not only enhances the scope of understanding but also improves the richness of the model's interpretation, providing a more comprehensive view of the image. The positive impact on all evaluative measures reaffirms the effectiveness of the NLGM. Therefore, capturing these long-range dependencies through non-local guidance is an essential element for improving the performance of our proposed method on both DUTS-TE and DUT-OMRON datasets.

3.6.6.2 Effectiveness of FFG

The incorporation of feature fusion, as depicted in the 4th and 6th rows of Table 3-3, additionally boosts the performance of our model, surpassing the results of configurations without the Feature Fusion Gate (FFG). This improvement is evident across all evaluative metrics on both DUTS-TE and DUT-OMRON datasets. This demonstrates that feature fusion gate plays an important role in our posed method.

Feature fusion gate's role in the architecture is like a gatekeeper, gradually sifting the input data, and ensuring that only the most relevant and significant features are used. This mechanism results in a substantial reduction of noise and unnecessary information, thereby enhancing the precision of our model. Integrating feature fusion into the system not only refines the process but also improve the performance to a higher level, outperforming models that don't incorporate the FFG. The results indicate the critical role of feature fusion in our method, demonstrating how effectively it removes redundant data, leaving only the most pertinent information in model's architecture.

3.6.6.3 Effectiveness of ERM

As indicated in Table 3-3, E denotes the output from the initial Edge-Refinement Block (ERB) layer. This output is then passed to various stages

of our decoder, in a process similar to that designed in the EGNet. The introduction of the Edge-Refinement Module (ERM) results in a performance enhancement, as seen by comparing the 5th and 6th rows in Table 3-3.

We hypothesize that the convolution operations of the ERM, applied in a cascade manner, assist the edge features in adaptively supporting the learning of non-local and salient features at various stages and resolutions.

3.6.6.4 Effectiveness of NLGM & ERM

When compared to the results from the second row, the third row in Table 3-3 shows an improvement in performance. This serves as evidence of the effectiveness of introducing both the NLGM and ERM into our approach. The features that offer mutual benefits to both non-local and edge features are emphasized, enhancing the overall performance.

These two modules seem to have a symbiotic relationship, each one benefiting and enhancing the performance of the other to create a comprehensive and boundary-aware model. The combination of non-local and edge allows for an accurate reconstruction of the salient regions in the image.

3.6.6.5 Effectiveness of FFG & ERM

The introduction of edge features embedded into the Feature Fusion Gate (FFG) shows noticeable performance improvements, as evidenced by the comparison between the 6th and 4th rows in Table 3-3. This indicates that the Edge-Refinement Module (ERM) plays a critical role in promoting relevant salient features before they are combined in the FFG.

The ERM works as a key first step for the features before they go into the FFG. It fine-tunes and sharpens the edges of saliency features, which helps important parts of the image stand out and helps the model to separate

interesting objects from their background. This early step is vital to the whole process as it provides a clear input for the FFG, making it easier for it to sort and mix the most important features.

In summary, adding edge details as a key part of the FFG bring performance gains, which demonsrates the benefit of using fine-tuned edge details to guide the mixing process, ultimately improving the final output of the model.

3.6.6.6 Configurations of NLGM

NLGM Configurations	DUTS-TE			DUT-OMRON		
	MaxF ↑	MAE ↓	S ↑	MaxF ↑	MAE ↓	S ↑
SSNLB	0.8813	0.0411	0.8789	0.8081	0.0563	0.8336
CSNLB	0.8816	0.0415	0.879	0.8079	0.0566	0.8348
SSNLB+CSNLB	**0.8836**	**0.041**	**0.8809**	**0.8103**	**0.0562**	**0.8357**

Table 3-4 Performance comparison of different NLGM configurations. SSNLB and CSNLB refer to spatial-space non-local block and channel-space non-local block respectively. All three configurations are without FFG and ERM.

Table 3-4 presents the results of our experiments designed to assess the performance of various configurations of the NLGM. When compared to the baseline model (presented in the 1st row of Table 3-3), the models that incorporate either the Spatial-Space Non-Local Block (SSNLB) (shown in the 1st row of Table 3-4) or the Channel-Space Non-Local Block (CSNLB) (displayed in the 2nd row of Table 3-4) both exhibit improved performance on the two datasets.

In comparison, the best performance is achieved when both the SSNLB and CSNLB are utilized together, as shown in the 3rd row of Table 3-4. This suggests that these two modules work complimentarily, each enhancing the other's contribution. The results indicate that both spatial and channel-wise

non-local features are crucial in highlighting the salient objects in an image. This combination of both modules allows the model to benefit from spatial space and channel space, making our model more robust in handling complicated MSOD scenarios.

3.6.6.7 Architectures of NLGM

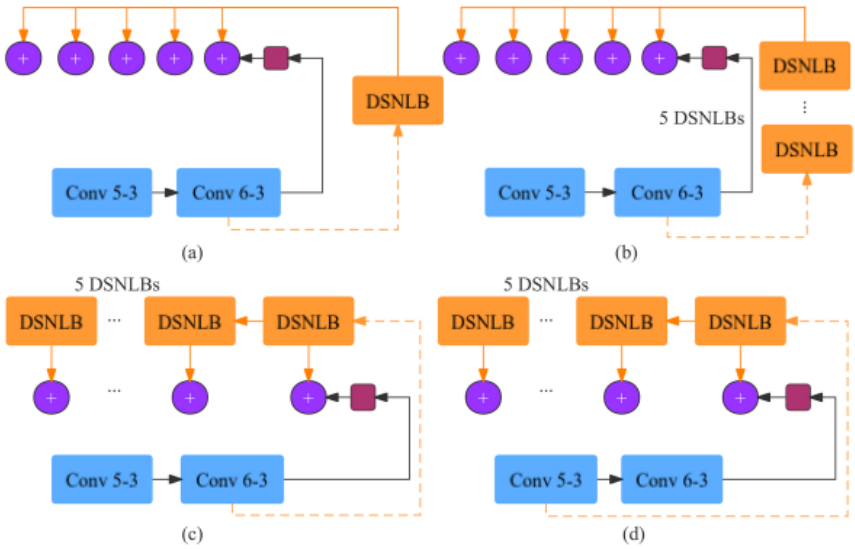


Figure 3-10 Different architectures of NLGM. All structures here are without FFG and ERGM. Element-wise addition operation is used at each stage to fuse different features.

NLGM Architectures	DUTS-TE			DUT-OMRON		
	MaxF ↑	MAE ↓	S ↑	MaxF ↑	MAE ↓	S ↑
(a)	0.8796	0.0419	0.8787	0.8052	0.0567	0.8313
(b)	0.8815	0.0415	0.8801	0.8075	0.0565	0.833
(c)	0.8816	0.0417	0.8805	0.8083	0.0565	0.8345
(d)	**0.8836**	**0.041**	**0.8809**	**0.8103**	**0.0562**	**0.8357**

Table 3-5 Performance comparison of different NLGM architectures. All structures here are without FFG and ERM.

We perform experiments to explore the effect of different structures of NLGM. We evaluate four different architectures to incorporate non-local information into our U-shape network. The architectures and the corresponding performance are shown in Figure 3-10 and Table 3-5 respectively.

Model (a) is the simplest version of our design. This baseline model employs a single DSNLB to draw features from the Conv6-3. The output from this is then distributed across all top-down stages. This model sets the benchmark for our following architecture experiments.

Model (b) is an extension of Model A. Instead of using a single DSNLB, we incorporate a sequence of 5 DSNLBs, all drawing from the same layer Conv6-3. This modification allows us to evaluate the impact of having chains of non-local blocks over single instances. From our findings, the stacked DSNLBs enhance all performance metrics, demonstrating the effectiveness of long-range multi-hop communications in building richer salient features, across both spatial and channel dimensions.

Model (c) is an experiment where the features are again sourced from Conv6-3, but the DSNLBs are spread out along the top-down pathway. This alignment follows our main architecture as depicted in Figure 3-4. The spread-out DSNLBs result in slightly improved performance, likely because this one-to-one guidance method is more capable of generating adaptive non-local features that are suitable for each saliency scale.

Model (d) is our final architecture, which draws features from Conv5-3 instead of Conv6-3. The larger spatial size of these features from Conv5-3 is potentially better utilized by the DSNLBs. This performance improvement implies better non-local features are exploited in this larger spatial size feature (Conv5-3) compared to the one using relatively small spatial dimensions of the features from Conv-6-3.

3.7 Conclusion

This chapter focuses on tackling the challenging task of segmenting multiple salient objects. We introduce a novel framework for MSOD, taking advantage of both spatial and channel-wise long-range dependencies.

A key component of our system, the Non-Local Guidance Module, is capable of capturing long-range dependencies among salient objects distributed across the image. This module significantly enhances the network's ability to distinctly identify multiple salient objects.

Furthermore, we propose a Feature Fusion Gate, an innovative module designed to merge both salient and non-local features. This gate employs progressively refined edge features, which aids in highlighting the most pertinent features extracted from each module, thereby providing a more detailed and comprehensive view of the scene.

Our method provides state-of-the-art performance across five widely used datasets, demonstrating its effectiveness and generalizability. Moreover, we have curated an additional dataset, specifically composed of scenes with multiple salient objects. A comparison with other methods reveals that the performance gap becomes even larger with this complex dataset, indicating the strengths of our approach in dealing with complicated multiple salient objects scenes.

We believe the MSOD is an important area that requires deeper exploration in order to drive further advancements in image saliency research. Our network, along with the newly curated dataset, can serve as a baseline for performance evaluation in this field.

It is acknowledged that the human visual system can simultaneously focus on multiple objects within a visual scene. Each of these salient objects captures our attention, contributing to the overall understanding of the scene. When studying saliency, it is therefore important to consider the existence of multiple salient objects and their interaction with one another, as this

reflects the complex human visual perception more accurately.

When a visual scene includes multiple salient objects, different salient objects may have different importance. It is not sufficient to merely detect these multiple salient objects; understanding their relative importance is equally critical. As such, from the next chapter, we extend the problem and examine the complex issue of RSR. We aim to assign relative ranking to multiple salient objects within a single image, effectively distinguishing not just the objects themselves, but their relative significance within the image context.

Chapter 4 Saliency Ranking Dataset

4.1 Introduction

Chapter 3 introduced a method for SOD. We utilized non-local blocks to increase the ability of the network to combine features of a longer distance in an image, so-called long-range dependencies. We curated a new dataset to test this ability and found our approach to offer strong performance compared with recent methods. The problem solved was MSOD, which is inherently a binary segmentation task. In practice, for multiple salient object scenes, it can be challenging to determine which objects are or are not salient. Or, for example, which object is the most salient. New datasets and techniques have begun to be developed to address this, with the problem of RSR. New datasets in this area, however, often have limited number of objects making performance on complex scenes less effective. These datasets have typically been created based on mouse-trajectory data, rather than real human visual attention. The objective of the work in this chapter is to establish a large-scale dataset specifically for the instance-level RSR. We create a novel dataset of images in which object saliency is measured based on human eye tracking data. In doing so, we produce a challenging and diverse dataset, which can be used to develop new techniques for salient object ranking.

Section 4.2 will first explore the current saliency ranking datasets. The research gaps will be outlined in Section 4.3. Our data collecting strategy used to create the proposed dataset is illustrated in Section 4.4. The data structure and examples of proposed dataset are shown in Section 4.5. Finally, the statistics of the proposed dataset will be shown in Section 4.6.

4.2 Background

RSDNet [82], as a pioneering work in RSR, is trained and evaluated on the PASCAL-S [77] dataset. However, this dataset is suboptimal for this task due to several reasons. Firstly, it comprises only 850 images, with 40.4% of them containing just one saliency rank. Such images are unsuitable for training and evaluating saliency ranking models, while the remaining images are insufficient to train deep models effectively. Secondly, numerous images in this dataset present following problems: Multiple instances within the same image are annotated with the same rank. Some instances are over-segmented into multiple regions with varying rank values. Both of these situations are inappropriate for the instance-level saliency ranking detection task. The researchers of [82] extend this idea and combine MS-COCO dataset [79] and SALICON dataset [80] in the journal version of their paper [81]. To be more precise, MS-COCO contains intricate images with object segmentation, while SALICON is an extension of MS-COCO designed specifically to provide mouse-trajectory-based fixations. Within the SALICON dataset, fixation data is available from two sources: 1) sequences of fixation points and 2) fixation maps corresponding to each image.

Figure 4-1 An example of the mouse-contingent stimuli proposed in SALICON dataset [80]. The red circles indicate the movement of mouse cursor from one object to another.

Figure 4-1 demonstrates an example of the fixation collecting strategy in SALICON dataset. In SALICON dataset, all the images are firstly blurred.

Each image is displayed for 5 seconds, followed by a 2-second waiting interval. The mouse cursor is shown as a red circle with a radius of 2 degrees of visual field, allowing for free viewing of the high-resolution focus area. The cursor automatically moves to the image center when the image appeared. The subjects are given the freedom to explore the images by moving the mouse cursor to any location they wish to look at. No specific instructions are provided on how to move the mouse or where to direct their gaze in the images. Whenever the subjects move the mouse, the display updates accordingly, with the high resolution area centered on the mouse position. The mouse position and the timestamp are recorded throughout the experiment.

The authors [81] employ complicated hand-designed rules with adjustable parameters to curate their datasets by filtering out unsuitable images and selecting salient object instances. They create two versions of the dataset, one with loose parameters, resulting in a dataset, and another with stricter parameters, yielding a cleaner dataset. Despite their careful tuning and verification on smaller-scale data, these complicated designed rules may not be suitable for all scenarios and be able to guarantee annotation accuracy, particularly in complex visual scenes.

4.2.1 ASSR Dataset

To solve the RSR problem, Siris et al., 2020 [78] propose a large-scale salient object ranking dataset (ASSR) based on the combination of the MS-COCO [79] and SALICON [80] datasets.

In ASSR dataset, the key idea of assigning rank is related to attention shift, where up to the 10 object polygons in MS-COCO are considered, but for saliency ranking ground truth, top-5 objects are set to be a limit. ASSR concentrates on distinct objects fixated in a sequence while disregarding any repeated objects. In this process, the researchers here assign descending

scores to objects based on their order of fixation and then average these scores across all observers. In essence, the higher the score of an object, the higher its rank in the saliency list.

The segmentation annotations of ASSR dataset are drawn from MS-COCO, and the ranking information is captured from the mouse-trajectory-based fixations in SALICON dataset, focusing on the sequence of these simulated fixations. ASSR dataset directly discards the images without object annotations in MS-COCO dataset and the images with smaller objects enclosed by large objects. The images containing at least two salient objects are selected to make sure all the images have relative ranking information. Finally, the ASSR dataset is created with 7646 training images,1436 validation images and 2418 test images.

4.2.2 IRSR Dataset

IRSR dataset [110] is also created based on the combination of MS-COCO dataset and SALICON dataset. Several challenges are proposed in this work while combining the two datasets. Firstly, MS-COCO contains only 80 annotated classes of object instances, leading to instances in some images having sufficient fixations but lacking mask annotations. The second challenge is related to the large number of annotated instances in many COCO images, which cannot all be utilized for saliency ranking due to the subjective nature of saliency perception. Humans may struggle to rank the saliency of numerous objects, especially those with a lower degree of saliency. Consequently, it becomes necessary to carefully select salient instances among the complex background while disregarding non-salient ones. The third challenge is related to the presence of annotation errors in MSCOCO, such as images that are either over-segmented or under-segmented. These various challenges collectively hinder the direct utilization of these two datasets for saliency ranking purposes.

The researchers here argue that ASSR dataset is not optimal since it applies a straightforward filtering approach by removing images without object annotations and those containing smaller objects entirely enclosed by larger ones. Although this method helps select images, it also introduces noise, particularly because they are unable to filter out images with salient objects falling outside the 80 MS-COCO classes. Furthermore, they generate saliency ranking annotations based on attention shift, which results in their dataset being more similar to a scanpath prediction task rather than the RSR task.

In this study, a more accurately annotated dataset is constructed for the relative salient instance ranking task. The process involves selecting 15,000 images from the SALICON dataset within MS-COCO and extracting their instance segmentation masks. Subsequently, these images, along with their instance annotations, are presented to different subjects, consisting of five postgraduate students aged between 20 to 30, with four males and one female. The subjects are tasked with identifying appropriate images and selecting salient objects.

The selection process obeys specific rules:

(1) Each subject individually selects the objects they think to be salient within each displayed image.

(2) Images containing salient objects that are not annotated in the MS-COCO instance annotations are marked as inappropriate.

(3) Images with evident segmentation errors, such as instances that are over or under-segmented, are also deemed inappropriate.

(4) Images with more than eight or fewer than two salient instances, or lacking clear salient objects, are considered unsuitable. The maximum number of salient instances is limited to eight, following the PASCAL-S dataset, which has at most seven saliency ranks in each image, considering the relatively large number of objects in MS-COCO images.

Following the manual selection and annotation process, images marked as inappropriate by more than three subjects are filtered out. For the remaining images, objects labeled as salient by at least three subjects are designated as salient instances.

To establish saliency ranking within each image, the researchers use the saliency maps provided by the SALICON dataset instead of fixation points. However, instead of utilizing the average saliency value within each instance mask, the labeled salient objects are ranked based on the maximum saliency value within their respective instance masks, as the degree of saliency for an object is primarily determined by its distinctive parts.

The final dataset comprises 8,988 images, divided into 6,059 training images and 2,929 test images, following the training and validation split of SALICON. Similar to the PASCAL-S dataset, both instance segmentation and relative saliency ranks are represented as saliency maps. However, unlike PASCAL-S, saliency values in different salient instance masks are assigned by uniformly dividing the range [0, 255] based on their saliency rank orders.

4.2.3 Summary of Saliency Ranking Dataset

The summary of two popular saliency ranking datasets is presented in Table 4-1. The ASSR dataset emphasizes the sequence of attention shifts in saliency ranking, while the IRSR dataset focuses on a more manual and detailed process of selecting and annotating images, aiming for a more accurate representation of relative saliency. Both datasets aim to provide a more robust foundation for training and evaluating RSR models.

Feature	ASSR Dataset	IRSR Dataset
Base Datasets	Combination of MS-COCO and SALICON	Combination of MS-COCO and SALICON
Rank Assignment	Based on attention shift, up to top-10 objects considered, but top-5 objects for saliency ranking	Ranking based on the maximum saliency value within instance masks
Image Selection	Discards images without object annotations in MS-COCO; images with at least two salient objects selected	Manual selection by subjects; inappropriate images and those with segmentation errors removed
Segmentation Annotations	From MS-COCO	From MS-COCO
Ranking Information Source	Mouse-trajectory-based fixations from SALICON	Saliency maps from SALICON, objects ranked by maximum saliency value within masks
Dataset Size	7646 training, 1436 validation, 2418 test images	6059 training, 2929 test images
Challenges Addressed	Filters out images with smaller objects enclosed by larger ones; focuses on sequence of fixations	Addresses issues of subjective saliency perception, annotation errors in MS-COCO; limits salient instances to 8 per image
Methodology	Assigns descending scores based on order of fixation; averages scores across observers	Objects labeled as salient by at least three subjects are designated as salient; ranks based on the maximum saliency value

Table 4-1 Summary and comparison of current popular saliency ranking datasets ASSR and IRSR.

4.3 Research Gaps

4.3.1 From Mouse-Trajectory based Fixations to Eye-Tracker based Fixations

The creation of both the ASSR and IRSR datasets is focused on instance-level salient object ranking, achieved through the combination of the MS-COCO dataset and the SALICON dataset. However, the data in the SALICON dataset is composed of mouse-trajectory based fixations. Eyes

and the visual cortex are fundamentally different from the parts of the brain that control hands to move a mouse. The are several issues while imitating the human beings' gaze using mouse-tracking strategy:

Accuracy: Fixations collected using mouse-trajectory-based methods may not be as precise as real human eye gaze data. Eye-tracking technology provides more accurate and fine-grained information about gaze positions and durations, while mouse-tracked fixations may be influenced by hand-eye coordination and operational errors, leading to estimation biases in fixation locations and durations.

Physiological Features: Eye-tracking data can reveal physiological features and cognitive processes subconsciously, such as involuntary changes in eye movement patterns. This information is crucial for understanding visual attention mechanisms and cognitive processes. The mouse-trajectory-based data in the SALICON dataset resembles more of a human's exploration path based on their interests. When presented with a blurry image on the screen, individuals consciously move the mouse to their point of interest. Therefore, SALICON dataset is really examining what people consciously find interesting, which is inherently a different task.

The dataset proposed in this chapter introduces a significant advantage by incorporating real human eye-tracking data, capturing the genuine gaze of human observers. By using eye tracker data, we are able to obtain more reliable and accurate information about the saliency of objects within the images, reflecting the genuine perceptual responses of human viewers.

The inclusion of real human gaze in our proposed dataset enhances the credibility and validity of the annotations, making it more suitable for training and evaluating models that are intended to capture and predict human visual attention. With this approach, we aim to provide a more realistic and reliable dataset for researchers and practitioners working in the field of computer vision, image processing, and visual attention modeling.

Figure 4-2 demonstrates some common images in all three datasets. As

can be seen, given same image, the salient objects and corresponding ranking order generated from mouse-trajectory fixation is different from our real fixation based dataset. The difference in ranking is because of different ranking strategies and two totally different modalities: mouse-trajectory fixation and real fixation.

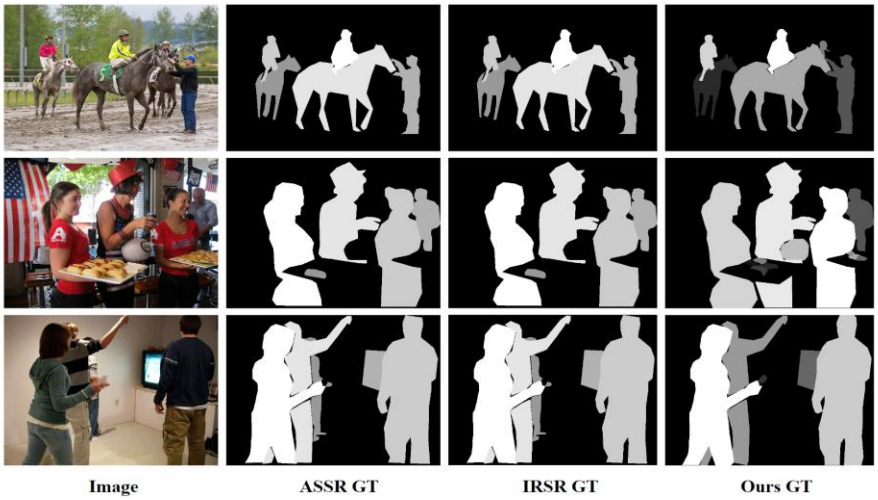


Figure 4-2 Some common images in all three datasets.

4.3.2 From Fixed Number Salient Instances to Unlimited Instances

The ASSR dataset imposes a restriction on the number of salient instance rankings, limiting it to 5, whereas the IRSR dataset allows a maximum ranking of 8. In our proposed dataset, we have chosen not to limit the number of salient object rankings. The rationale behind this decision lies in the inherent nature of human visual perception.

Human beings have the remarkable ability to visually focus on multiple salient objects simultaneously. In real-world scenarios, our attention can effortlessly shift between numerous points of interest, each carrying its own level of significance. Setting an arbitrary limit on the number of ranked salient objects in the dataset could potentially constrain the modeling of this inherent human capability.

By allowing for an unrestricted number of salient object rankings, our

dataset aims to better reflect the complexity and diversity of real visual attention processes. This provides a more comprehensive and representative dataset. Removing such constraints also aligns with the goal of developing more robust and adaptable models that can handle a diverse range of saliency scenarios.

4.3.3 Image Selection and Instance Ranking Annotations

Both ASSR and IRSR combine MS-COCO and SALICON datasets, where the former provides the instance polygons and the latter gives mouse-trajectory fixations.

ASSR employs a straightforward filtering approach to combine the MS-COCO and SALICON datasets, which directly removes images lacking object annotations in MS-COCO dataset, as well as those containing smaller objects fully enclosed by larger ones. This simple method introduces noise between the human mouse-trajectory fixations in SALICON, and the eventual ground truth in the ASSR dataset. The generated saliency ranking in ASSR is based on the attention shift, which is close to a scanpath prediction task rather than a saliency ranking task.

The creation process of IRSR dataset directly shows each image with instance polygon annotations to different participants, for them to select if the objects are salient or not. This approach may introduce a potential bias for the participants; When instance annotations are visible to the participants, it can unfairly influence their judgments, particularly for objects that lack annotations. The presence of annotations, such as bounding boxes or polygons, draws attention to specific regions and shapes within the image. Consequently, participants may be inadvertently influenced by these annotations when making their saliency determinations. In such a setup, the participants' visual system might be subtly guided towards objects with

explicit annotations, while potentially overlooking equally salient objects that lack such annotations. This can result in an unintended skew in the dataset, impacting the overall accuracy and reliability of the human judgments. To ensure a more fair and objective evaluation of saliency, it is crucial to minimize any external factors that could bias the participants' perceptions.

4.4 Data Collecting Strategy

4.4.1 Step 1: Image Selection

The objective of this work is to establish a large-scale dataset specifically for the instance-level RSR. To meet this objective, our selection criteria for images is limited to those sourced from the MS-COCO dataset containing more than three instances. We argue that images with fewer than three distinct instances might not provide enough variation in saliency to be informative for the study. By setting such a threshold, we aim to ensure a sufficiently complex visual environment, thereby facilitating a more comprehensive examination of the relative saliency ranking of different instances within a single image. After an automatic threshold, 4 subjects (2 males and 2 females) who were told the purpose of this dataset are asked to view these images without original annotations. For each image shown, each subject selects if the image is appropriate according to following criteria:

(1) If the image does not have more than 3 clear disconnected instances, the image will be marked as inappropriate.
(2) The background area or objects will not be considered as one of the minimum three instances required for image selection.

After the process, any images that are marked as inappropriate by more than two subjects will be filtered out from the dataset.

4.4.2 Step 2: Gaze Recording

After getting the initial dataset, we proceed with a task of free viewing guided by an eye-tracking system, which is conducted by eight subjects (the authors of this work are exclusive). This group is balanced in terms of gender, consisting of four males and four females, all of whom are within the age range of 20 to 30 years. The tracking of gaze data is performed using a Tobii Pro Nano eye-tracker, set to a sampling frequency of 60 Hz.

In order to ensure the fairness and consistency of the data collection process, all subjects interact with identical hardware. Specifically, they use the same eye-tracking device, the same 23-inch Lenovo ThinkVision T2364PA monitor with a resolution of 1920x1080 pixels and a refresh rate of 60 Hz, and the same PC. These devices are utilized sequentially by the subjects to ensure the uniformity of the viewing experience and the fairness of data.

Each image is firstly resized proportionally to fit the full screen of 1920 x 1080 resolution, which is then presented to the subjects for a fixed duration of 3 seconds. To maintain accuracy and calibration of the eye-tracking device, a recalibration procedure is carried out every 200 images. Alongside this, to relieve eye fatigue and ensure the quality of data, participants are offered the opportunity to rest or pause the gaze recording process at these same 200-image intervals.

It should be noted that this procedure is not a swift one, the entire process of gaze recording is spread over a span of six months, during which time all eight subjects complete their tasks. This deliberate pace is maintained to ensure the accuracy and reliability of the data captured, which in turn contribute robustness to this dataset.

4.4.3 Step 3: Fixation Filtering

The Tobii Pro Nano eye tracker can record different kinds of gaze information, such as recording timestamp, gaze point location (x and y), presented media name, presented media width, presented media height, original media width, original media height and so on. This information is utilized to filter out the fixation point.

The term filter is frequently used in eye-tracking analysis software and here it denotes the steps designed to identify the fixations with the raw gaze data. In eye-tracking research, two commonly used words to classify these data are “saccade” and “fixation” [84]. A saccade is a quick movement of the eyes that occurs between two fixation points. These are extremely fast jumps that the eyes make to move from one point of interest to another. Despite the fact that the eyes are moving during a saccade, the brain typically doesn't process visual information during this period, a phenomenon known as saccadic suppression [128]. In comparison, a fixation is a period when the eyes are still and focused on a particular point. During this time, our eyes are gathering and processing information about the object we are looking at. In the phase of fixation, our gaze is relatively stable, allowing our brain to process and interpret the image we're seeing. Within the field of human behavior research, fixations often gather the most interest in studies of eye movement. This is primarily because fixations provide indications of when and what information the brain registered during the visual observation process. For the problem of RSR, we think the duration of fixation points will directly reflect the relative ranking information among different instances. Therefore, it is necessary to filter out the fixations in the gaze raw data.

Velocity-Threshold Identification (I-VT) [85] is an algorithm utilized for fixation classification, and its function relies primarily on velocity-based measurements. The core concept of the I-VT filter lies in the classification

of eye movements according to the velocity of the eye's directional shifts. When the calculated velocity exceeds a certain threshold, the corresponding sample is classified as a saccade; conversely, if the velocity falls below the threshold, the sample is identified as part of a fixation.

Following the velocity-based fixation classification methods, the simple yet effective way is utilized to calculate the velocity among different gaze data points (see Equation 4-1):

$$V_{t1t2} = \frac{|S_{t1} - S_{t2}|}{|t_1 - t_2|} \qquad (4-1)$$

Where the Euclidean distance between two consecutive samples is calculated and then divided by the sampling time.

We then set the threshold. Our saccade threshold is experimentally set to 1.5 pixels per millisecond, which is 1500 pixels per second. The gaze data and velocity chart from one subject while looking at an image in 3 seconds are shown in Figure 4-3. The blue line indicates the gaze point position along the x-axis, while the orange line indicates the calculated velocity between every two consecutive gaze points. The red indicates the saccade threshold based on the calculated velocity. In the classification process, each sample undergoes a velocity check, determining its classification as either a fixation or a saccade. When the velocity is below velocity threshold (red line), it is categorized as part of a fixation. Conversely, if the velocity meets or exceeds this parameter, the sample is classified as part of a saccade. The duration of fixation here can be calculated in the gap of two saccade.

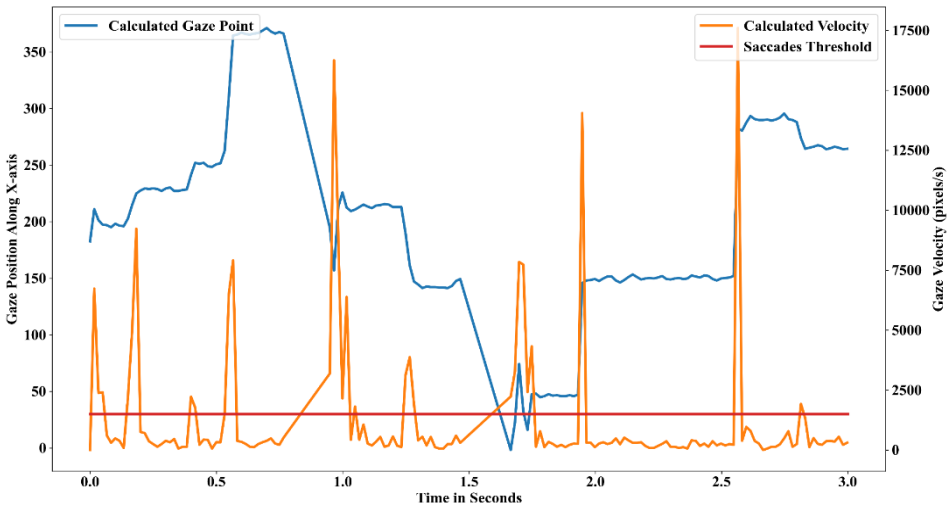


Figure 4-3 Gaze data and velocity chart from one subject while looking at the image above. Here, 9 effective fixation events are captured after the threshold.

After this, any fixation points exceeding a duration of 200 milliseconds are selected as effective fixation event. This decision is inspired by the research proposed in [84], which indicates that during periods of fixation - moments when our eyes remain still between saccades - the duration tends to range from approximately 200 to 300 milliseconds. Note that to reduce the center bias, we remove the first fixation point of each subject in each image.

Finally, all images will be resized to the original image size and the corresponding effective fixation points in fixation event will be relocated proportionally. Then, all the noise fixation points out of the bound of the image size will be deleted (see Figure 4-4).

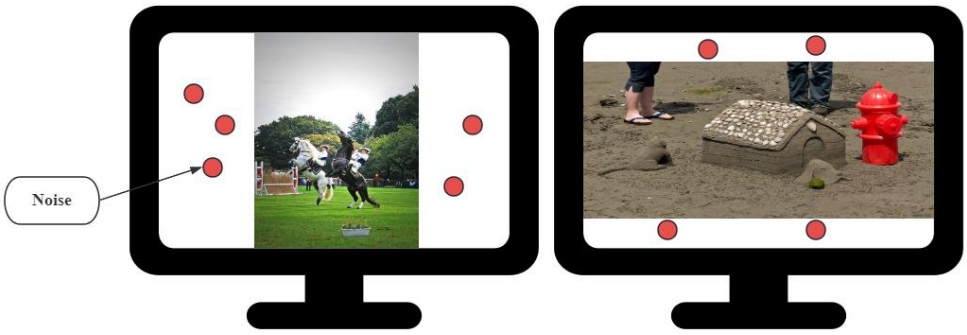


Figure 4-4 Examples of noise fixation points in gaze recording.

4.4.4 Step 4: Salient Objects Threshold

In order to generate the relative saliency ranking at instance level, it is necessary to firstly judge if the corresponding instances are salient objects. It is reasonable to combine the fixation data and the polygons information in MS-COCO for this task. A threshold is set to filter the salient instances based on the number of fixation points in each polygon.

The complexity of this task arises while utilizing the MS-COCO dataset, which although provides instance-level polygon data, it does not offer an adequate quantity of polygons for each image to satisfactorily fulfill the needs of our task. We often encounter scenarios where fixation points exist within a object, however, the corresponding polygon information for this instance is not incorporated within the available polygons from the MS-COCO dataset. Meanwhile, some images selected from MS-COCO test set do not have any annotations. Such instances present obstacles in our study. Second, MS-COCO dataset sometimes provides big polygons for a group of crowded instances, which cannot be directly used. To address the aforementioned problem, several strategies are applied (see Figure 4-5).

Note the following steps will generate 2 things for each image: (1) The image with different color fixation points representing different types of fixations. (2) The initial rough polygons of salient instances from MS-COCO dataset or Mask RCNN, which will be displayed in the next step for participants to annotate.

(1) Fixation points in existing standard MS-COCO polygons:

We calculate the mean number m of fixation points lying in different standard polygons and the corresponding standard deviation σ.

For the fixation points in existing MS-COCO polygons, if the existing MS-COCO polygon is not a crowded polygon, we do the following threshold:

$$N_{fixation} \geq m - \sigma \qquad (4-2)$$

Where $N_{fixation}$ is the number of fixation points in this polygon. $N_{fixation}$ should be bigger than $m - \sigma$.

If the polygon satisfies the above threshold, the polygon will be regarded as a salient instance, then the polygon and corresponding fixation points will be added to our initial draft dataset. These fixation points will be marked in red in the original image.

(2) Fixation points in existing crowded MS-COCO polygons:

If the existing MS-COCO polygon is a crowded polygon, the number of fixation points cannot be directly calculated here as the crowded polygon normally contains more than 2 instances. Therefore, these fixation points are input into the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [86] clustering algorithm. DBSCAN is a density-based clustering algorithm, as opposed to centroid-based methods like K-means [87]. DBSCAN operates under the principle that a cluster is a high-density area surrounded by a lower density region in the data space. The algorithm works by defining a neighborhood around a data point, and if there are enough points within this neighborhood, the data point is labeled as a core point and forms a cluster or part of a cluster.

The process on DBSCAN is illustrated as follows:

It starts with an arbitrary starting point that has not been visited. The neighborhood of this point is extracted using a distance epsilon (ε). If there are a sufficient number of points (according to a $minPts$ parameter) within this neighborhood, a new cluster is started. Otherwise, the point is labeled as noise, meaning it doesn't belong to any cluster. It's important to note that this point might later be found in a sufficiently large ε-environment of a different point and, thus, be made part of a cluster. If a point is found to be a dense part of a cluster, its ε-neighborhood is also part of that cluster. Hence, all points that are found within the ε-neighborhood are added, along with their own ε-neighborhood when they are also dense. This process continues until the

density-connected cluster is completely found. Finally, the process restarts with a new unvisited point.

The DBSCAN algorithm was selected in this experiment because it is a density-based algorithm without the need of providing the number of clusters. Here, DBSCAN algorithm is set with the ε and the $minPts$ being 20 and 4 (half of the number of gaze recoding participants) respectively. This will generate 2 different groups of fixation points, the points constructing a cluster and the ones not constructing a cluster.

After the clustering, only the fixation points constructing a cluster will be input to the same threshold as illustrated in (4-2): $N_{fixation} \geq m - \sigma$. The cluster passing the threshold are regarded as salient instance and the corresponding fixation points are then be added to initial draft dataset. The fixation points here will be marked as pink in the original image and at the same time, the bounding box for this kind of crowded polygon will be marked in original image.

The existing MS-COCO polygons can cover most of the situations in our experiment. However, there are still some situations that the recorded fixation points lie in some instances that are not annotated by MS-COCO dataset.

(3) Fixation points not in existing MS-COCO polygons:

For the images in MS-COCO test set and the fixation points not in existing MS-COCO polygons, DBSCAN clustering algorithm with the same parameters (ε and the $minPts$ being 20 and 4 respectively) is also applied.

If the fixation points do not construct a cluster, these points will be regarded as noise. If some fixation points construct one of the clusters, a threshold will be applied here $N_{fixation} \geq m - \sigma$, which is the same as the one used for thresholding the fixation points in normal polygons. The cluster passing the threshold will be regarded as salient instance and the corresponding fixation points will be marked as green in original image

and added to our initial draft dataset. Otherwise, if the points are not satisfying the threshold, these points will be marked as yellow in original image and these clusters will not be regarded as salient instance.

To reduce the workload of our annotators in the following steps, Mask RCNN [88] is firstly utilized to generate rough polygons. If the green points lie in the generated rough polygon, the corresponding polygon will be added to our initial draft dataset.

In summary, the above-mentioned strategies will generate 2 things for each image: (1) The image with different color fixation points indicating different types of fixations. (2) The initial rough polygons of salient instances from MS-COCO dataset or Mask RCNN, which will be displayed in the next step for participants to annotate. Note that we are not using Mask RCNN in crowded situation to generate rough polygons because normally the density of instances is very high in this situation and the Mask RCNN cannot handle this complicated scenario.

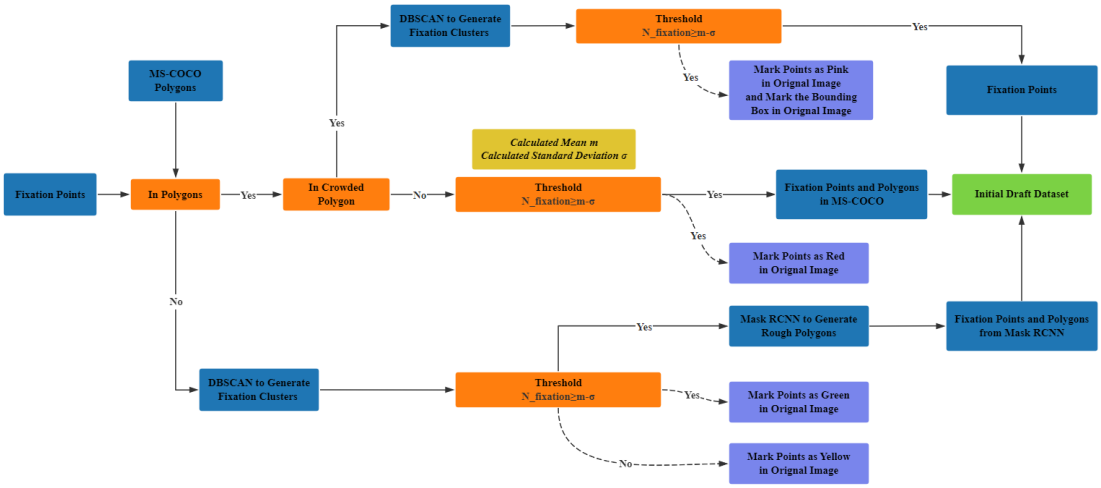


Figure 4-5 Several strategies applied to construct initial draft dataset.

4.4.5 Step 5: Annotating

Before illustrating the annotating process, two examples are shown to demonstrate the result from the initial dataset.

Figure 4-6 demonstrates an example without the crowded polygons. In this image, several fixation points have been shown in different colors. Red fixation points indicate the ones included in the existing MS-COCO polygons and these polygons are considered as salient instances. Green and yellow fixation points describe the ones not included in the existing MS-COCO polygons. Green fixation points denote the ones that have been classified as clusters and pass the threshold to be regarded as salient instances, and yellow fixation points denote the ones constructing a cluster but not pass the threshold, which cannot be regarded as salient instance. There are two polygons in the original MS-COCO dataset and two polygons generated from Mask RCNN here (always shown in the light pink). It can be found that there still some green clusters lie in an instance but there are no existing polygons for that.

Figure 4-7 shows an example with the crowded scene. In this image, green fixation points again represent clusters that pass the threshold (salient clusters), while the yellow fixations also represent clusters that do not pass the threshold (non-salient clusters). Red fixation points lying inside existing MS-COCO polygons and these polygons are regarded as salient polygons. It should be noted that there is crowded polygon here, where the pink fixation points indicate the ones lying inside the big, crowded polygon from MS-COCO dataset. Only the bounding box of the crowded area has been shown.

During the annotation process, we have ten participants, all ranging in age from 20 to 30, who are involved in the labeling task. The annotating process is based on LabelMe software, which is an open-source annotation tool. All the participants were instructed in the use of the software and were told the objectives of the study. Participants are asked to carry out two kinds of labelling tasks: (1) Create or refine the polygons (2) Assign classes to the polygons.

As the creation of this dataset is specifically for the RSR task. To mitigate the workload of the participants involved in the annotation process,

12 super classes out of the 80 available in the MS-COCO dataset are focused on strategically.

Each class out of the 80 available classes in the MS-COCO dataset is categorized under a super class, with a total of 12 such super classes. These super classes include 'person', 'vehicle', 'outdoor', 'animal', 'accessory', 'sports', 'kitchen', 'food', 'furniture', 'electronic', 'appliance', and 'indoor'.

By focusing on these super classes, the complexity of the task for the participants is reduced, thereby making the annotation process more manageable. Concurrently, the 12 super classes offer sufficient variety to ensure a comprehensive and diverse dataset for the RSR task.

Figure 4-6 Example 1: an image for participants to annotate in LabelMe software without crowded polygons.

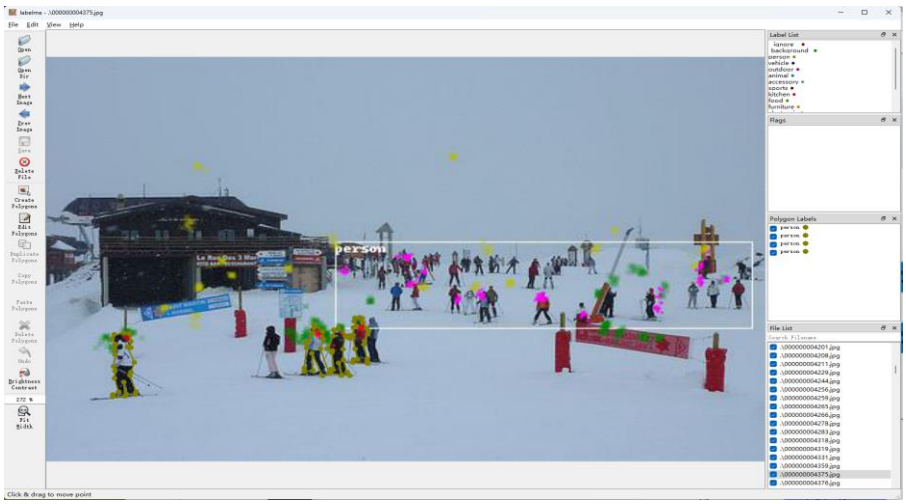


Figure 4-7 Example 2: an image for participants to annotate in LabelMe software with crowded situation.

In the process of creating or refining the polygons, for each image, the participants are following these rules:

(1) Check the existing polygons from MS-COCO dataset, refine all these polygons in instance-level with clear boundary and assign super class for the instances.
(2) If there is polygon generated from Mask RCNN, refine the polygon with clear boundary and assign the super class for the instance.
(3) If there is crowded scene containing a big bounding box, create instance-level polygons for the instances where the pink fixations points lie in and assign super class.
(4) Check the green fixation clusters, if there are no existing polygons, create instance-level polygons for the green fixation clusters and assign super classes.
(5) Check if there are 3 or more instance-level polygons, if the number of polygons is less than 3, mark the image as inappropriate.

Finally, two other subjects checked all images and checked the ones marked as inappropriate for deletion. After this step, 8389 images are annotated. 6701 images are chosen for training set in our dataset as these images are all from the MS-COCO training set and validation set, while 1688 images from MS-COCO test set are chosen to be the test set in our dataset.

4.4.6 Step 6: Ranking Assignments

The process of assigning ranks is based on the calculation of the number of fixation points within each polygon. These fixation points are derived from filtered fixation events with a duration exceeding 200 milliseconds. The number of fixation points within a polygon can reflect the duration of fixations, thereby providing an indication of the relative saliency importance information.

Upon determining the number of fixation points within each polygon,

the relative ranking ground truth can be generated. We represent ground truth ranking as sequentially increasing integers starting from 0, with higher values signifying a higher degree of saliency, and a value of 0 indicating the instance with the lowest degree of saliency.

It is worth noting that there is no predefined limit to the number of salient instances in each image for two reasons. Firstly, this choice is intended to enhance the complexity of the dataset for the RSR task, which we hope will provide a more accurate and comprehensive representation of the capabilities of our visual system. Secondly, setting an arbitrary limit on the number of salient instances within an image could potentially undermine the authenticity of the dataset and its ability to accurately represent the complexity of real-world visual perception. In real world, our eyes are constantly looking at different objects with a multitude of visual stimuli, and our brain is tasked with the complex job of identifying, categorizing, and prioritizing these stimuli based on their saliency. This process cannot be limited by an arbitrary limit, especially for the research of RSR.

However, for the other computer vision tasks and objectives, it is entirely feasible to establish a limit for the salient instances within our dataset. This flexibility allows for the dataset to be tailored to suit a variety of research purposes and experimental designs.

4.4.7 Step 7: Ground Truth Maps

Following IRSR dataset, ground truth maps are constructed based on the assigned rankings and the annotated polygons. Each individual polygon is allocated a distinct color, the formulation of which can be expressed as follows:

$$c_i = \frac{255}{L}(r_i + 1), \qquad (4-3)$$

Where c_i denotes the color of ith instance, L demonstrates the total instance number in this image and r_i indicates the ranking of ith instance.

Subsequently, the ground truth maps for the RSR task are generated. Notably, if different polygons are assigned the same color, this dataset can also be repurposed for the conventional SOD task.

As shown in Figure 4-8, it provides a visual representation of this process, demonstrating an image alongside its corresponding saliency ranking ground truth map and MSOD ground truth map. In terms of the Saliency Ranking Ground Truth (SR GT), it is evident that instance-level masks are generated with clear boundaries, with varying colors employed to indicate different degrees of saliency.

Given that each image in the proposed dataset encompasses three or more instances, this dataset can also make a significant contribution to the field of SOD. Specifically, it can serve as a challenging dataset especially for MSOD. The MSOD ground truth is clearly defined and of high quality, making it an ideal resource for tasks involving MSOD.

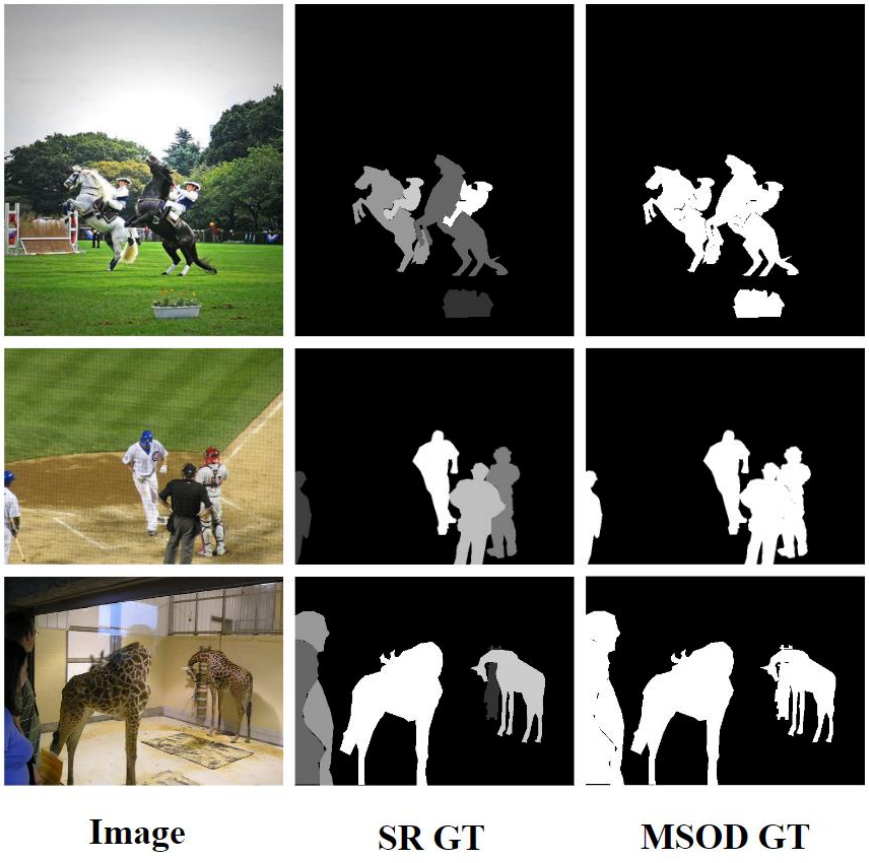


Figure 4-8 Examples of given images and the corresponding saliency ranking ground truth maps and MSOD ground truth maps.

4.5 Dataset Structure and Examples

The whole dataset is splitted into training set and testing set. Within each set, RGB images and corresponding ground truth maps are organized and stored in distinct directories. The associated annotations for these images are compiled and preserved in a JSON file.


```
{
  "file_name": "000000000063.jpg",
  "image_id": 63,
  "height": 480,
  "width": 640,
  "annotations": [
    {
      "instance": 0,
      "bbox": [77.0, 100.0, 205.0, 266.0],
      "segmentation": [list[1]],
      "category_id": 1,
      "gt_rank": 1
    },
    {
      "instance": 1,
      "bbox": [305.0, 110.0, 86.0, 216.0],
      "segmentation": [list[2]],
      "category_id": 1,
      "gt_rank": 2
    },
    {
      "instance": 2,
      "bbox": [52.0, 116.0, 34.0, 35.0],
      "segmentation": [list[1]],
      "category_id": 6,
      "gt_rank": 0
    }
  ]
}
```


Figure 4-9 An example of the annotations compiled in JSON file. Some JSON structure has been removed for clarity.

An illustrative example of an annotation for an image is provided in Figure 4-9. In this structure, the file name, image id, original image height, and original image width are provided. Following this, the annotations corresponding to different instances within the image are detailed. For each instance annotation, the bounding box is defined in an 'xywh' structure, which represents the position of the bounding box's center point, its width, and height, respectively. The 'segmentation' is a list that contains one or more sublists of points that construct the polygon. Occasionally, the list may contain more than one sublist, particularly when some instances are obscured by other instances and can only be represented by two or more separate polygons. Following this, the 'category_id' is displayed, which signifies the specific class of the polygon, such as person, animal, etc. Lastly, the ground truth rank information is provided.

In Figure 4-10, some examples of generating the proposed dataset have been shown. We present a series of illustrative examples that describe the process of generating the proposed dataset. For each image, the process starts with gaze recording, fixation filtering, and the salient objects threshold. These initial steps work for the generation of the fixation maps depicted in the second column. These fixation maps are subsequently presented to our participants for annotation. Following the annotation process, the polygons representing salient instances with distinct boundaries are presented in the third column. Subsequently, a ranking is assigned to each instance based on the number of fixation points contained within the polygons of different salient instances. The final step in this process is the generation of ground truth maps for RSR as shown in the fourth column. These maps provide a visual representation of the relative importance of different areas within the image.

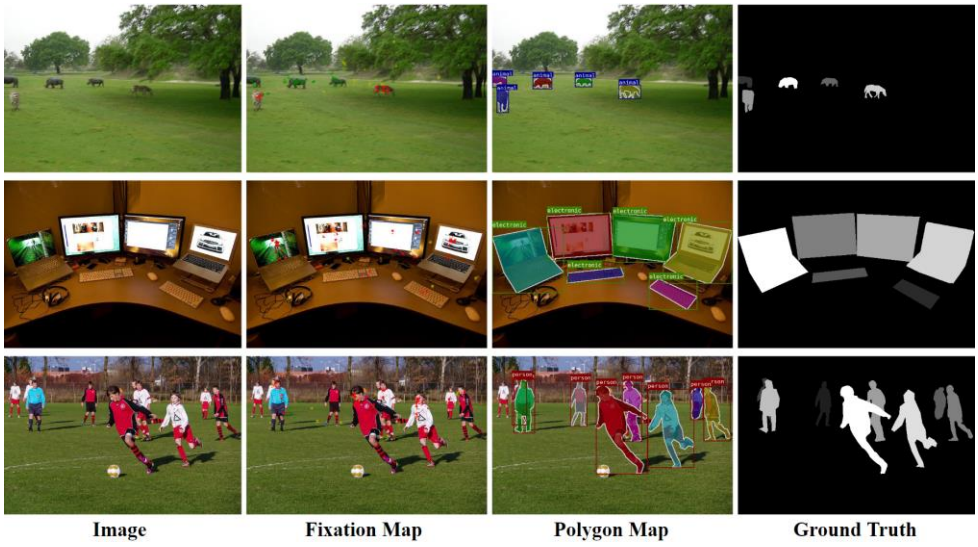


Figure 4-10 Examples of generated images in our dataset.

4.6 Statistics on Proposed Dataset

The dataset is composed of a total of 8389 images, with each image containing an average of approximately 6.22 salient instances. Out of these images, 6701 are allocated to the training set, while the remaining 1688 are assigned to the test set. The images in our training set are selected from the MS-COCO training and validation sets, while those in the test set are chosen from the MS-COCO test set. This allocation strategy is designed to better accommodate current transfer learning models. Considering the limit dataset in the area of RSR, most of the proposed models for saliency ranking utilize transfer learning techniques based on instance segmentation models, and the majority of these instance segmentation models are pre-trained on the MS-COCO training set. Therefore, it is reasonable to construct the test set in our proposed dataset exclusively from images in the MS-COCO test set. This approach ensures a fair benchmark for model performance evaluation.

The median number of salient instances per image in the dataset is 5. In total, the dataset encompasses 52173 salient instances, offering a rich and diverse collection for comprehensive analysis and study. This dataset is expected to provide valuable insights and contribute significantly to

advancements in the field of RSR.

4.6.1 Instance Number Per Image

Figure 4-11 demonstrate the statistics on the image quantity different salient instance numbers in each image. The dataset is curated to encompass a broad complexity level, as reflected by the number of salient instances per image ranging from the minimum number 3 to the maximum number 41. The distribution spans from relatively simple scenarios to highly intricate ones, thereby providing a comprehensive dataset for the development and evaluation of RSR models.

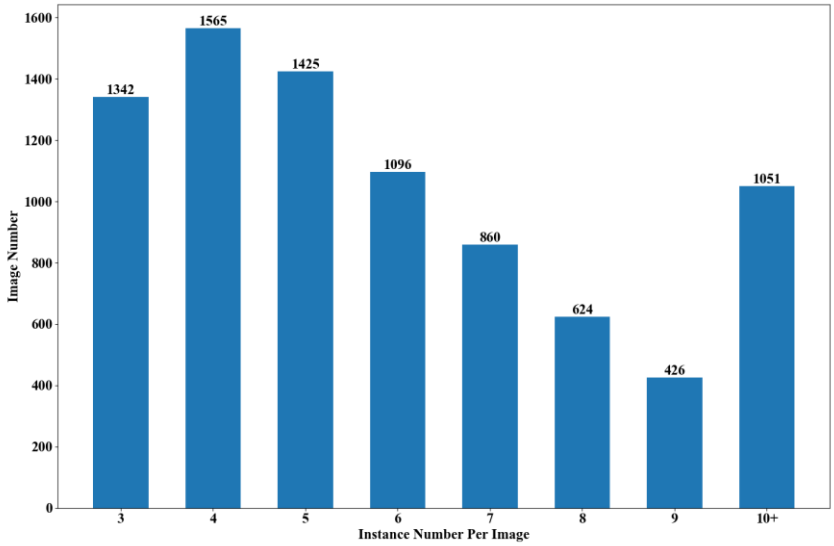


Figure 4-11 Statistics on the image quantity of different salient instance numbers in each image.

Approximately 16% of the images (1342 out of 8389) contain three salient instances. This subset of the dataset offers a substantial volume of simpler scenarios, where fewer objects are present, thereby facilitating the study of basic RSR tasks.

The most common scenarios in the dataset are represented by images containing four and five salient instances, accounting for approximately 18.66% (1565 images) and 16.99% (1425 images) of the total images,

respectively. These categories provide a rich source of data for investigating the dynamics of relative saliency ranking among a moderate number of instances.

As the complexity increases to images with six and seven salient instances, the count drops to 1096 and 860, respectively, accounting for approximately 13.1% and 10.3% of the total images. These categories present more complex scenarios, with a higher number of instances interacting within the same scene, thereby challenging the robustness of RSR models.

Further complexity is introduced with 624 images containing eight salient instances and 426 images with nine salient instances, representing approximately 7.4% and 5.1% of the total images, respectively. These categories, although less frequent, offer more intricate scenarios that push the boundaries of RSR tasks.

The dataset also includes 1051 images that contain more than ten salient instances each, representing the most complex scenarios and accounting for approximately 12.5% of the total images. These images are particularly valuable for testing the performance of RSR models under highly challenging conditions.

In summary, the dataset's diverse salient instance distribution, ranging from three to more than ten instances per image, ensures a comprehensive evaluation of RSR models. This variety not only enhances the models' adaptability but also their performance across a wide range of situations.

4.6.2 Instance Categories

The dataset includes a wide of instance categories shown in Figure 4-12, where the ‘Person’ and ‘Animal’ categories occupy the highest percentages, with the total proportion being approximately 74.3%. The remaining categories contribute to the remaining 25.7% of the salient instances. This

potentially indicates that observers may show more interest on the ‘Person’ and ‘Animal’ categories while doing the 'freeviewing' tasks.

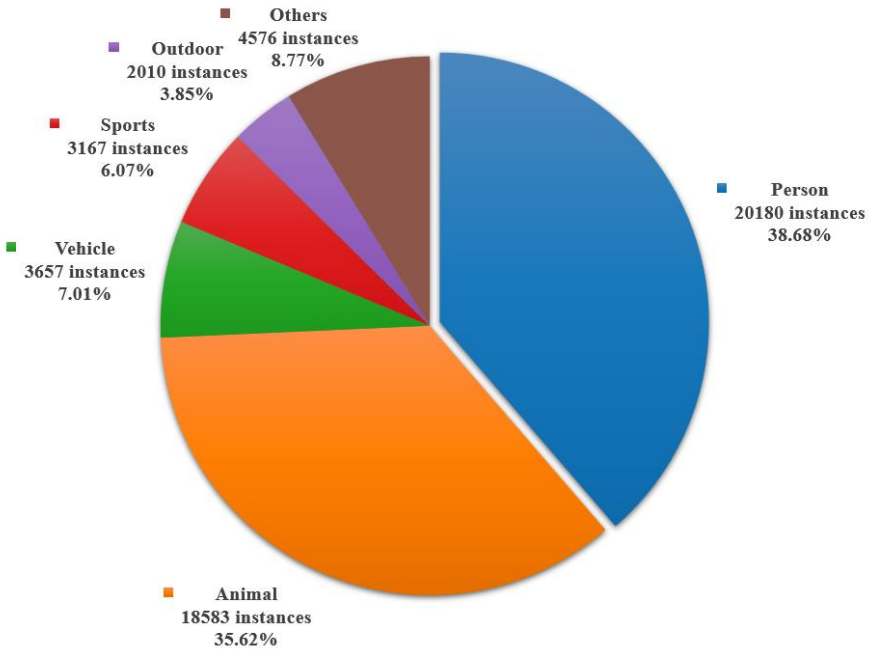


Figure 4-12 Statistics on the salient instances categories.

The 'Person' category, represented by 20180 salient instances, constitutes a significant portion of the dataset. This accounts for approximately 38.68% of the total instances, playing an important role of human-centric contexts in RSR tasks. The substantial representation of this category ensures that models trained on this dataset are proficient in handling scenarios involving human beings. The 'Animal' category, with 18583 salient instances, contributes to around 35.62% of the total salient instances, which ranks the 2nd. This category normally introduces a diverse range of shapes, sizes, and textures, thereby enriching the dataset and enhancing the robustness of the RSR models. The 'Vehicle' category, represented by 3657 instances, accounts for about 7% of the total salient instances and ranks the 3rd. This category introduces scenarios typically found in urban and transportation situations, ensuring that the models can effectively handle man-made salient instances and structures. The 'Sports' category, with 3167 instances, makes up approximately 6.1% of the total instances. This category

provides a variety of dynamic and high-speed scenarios, challenging the models to accurately identify the relative saliency regions. The 'Outdoor' category, represented by 2010 instances, accounts for nearly 3.9% of the total instances. This category introduces scenarios that involve the salient instances in outdoor environments, contributing to the diversity of the dataset. Lastly, the other categories ('accessory', 'kitchen', 'food', 'furniture', 'electronic', 'appliance', and 'indoor'), which includes 4576 instances, makes up around 8.8% of the total instances. These categories encompass a variety of salient instances and subjects not covered by the other categories, further enhancing the diversity of the dataset.

4.6.3 Relative Saliency Ranking in Categories

Figure 4-13 shows an average normalized saliency ranking in different categories across two datasets: our proposed dataset and ASSR dataset. Note that there is no category information provided in the IRSR dataset. To calculate the average normalized saliency ranking, first, for each instance within an image, we compute a normalized saliency rank. This is done by dividing the ground truth rank of the instance by the total number of instances within the image. Each calculated normalized saliency rank is then added to the list corresponding to its respective category. Finally, for each category, we compute the average normalized saliency rank. This is achieved by calculating the mean of all normalized saliency ranks within the category's list. This process results in an average normalized saliency rank for each of the 12 categories, providing a standardized measure of saliency across different categories.

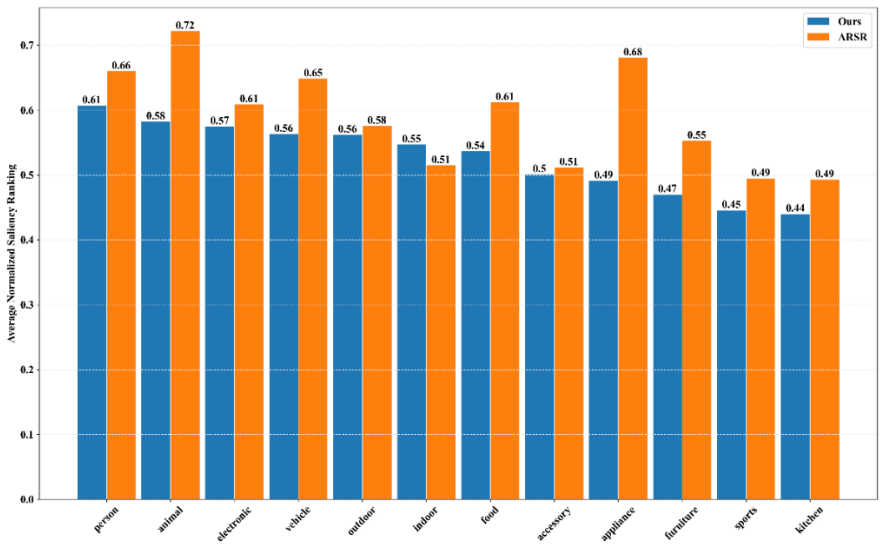


Figure 4-13 Statistics on average normalized saliency ranking in different categories. Blue bars indicate the data in our proposed dataset, while orange bars demonstrate the data in ASSR dataset. Note that ASSR dataset set a limit of salient instances to 5.

The blue bars in Figure 4-13 present the average normalized saliency ranks across 12 distinct categories in our proposed dataset, which provide insight into how attention is distributed across these categories within images. The category 'person' holds the highest average normalized saliency rank of 0.61, indicating that instances of this category tend to draw the most attention. This is followed closely by 'electronic' and 'animal' categories with saliency ranks of 0.57 and 0.58, respectively. On the other hand, the 'kitchen' category exhibits the lowest saliency score of 0.44, suggesting that instances in this category often own less attention relative to other instances within the same image. Other categories such as 'sports', 'furniture', ‘appliance’ and 'accessory' also have lower saliency scores, equal or under 0.5, indicating a generally lower degree of attention. The categories 'vehicle', 'outdoor', 'food', and 'indoor' present average normalized saliency ranks ranging from 0.54 to 0.56, indicating a moderate level of attention drawn towards these categories.

The ASSR dataset presents average normalized saliency ranks across the same 12 categories in orange. Note that ASSR is created based on the attention shift of imitated eye-movement as described in Section 4.2.1.

In ASSR dataset, the 'animal' category has the highest rank of 0.72, suggesting that instances of this category draw the most attention in terms of eye movement. This is closely followed by 'appliance' and 'person' categories with saliency ranks of 0.68 and 0.66 respectively. Conversely, the 'kitchen' and 'sports' categories have the lowest saliency scores, both at 0.49, indicating less eye movement towards these categories. Other categories such as 'indoor', 'furniture', and 'accessory' also exhibit lower saliency scores around 0.5 - 0.55. The categories 'vehicle', 'food', and 'electronic' have saliency ranks around 0.61 - 0.65, suggesting a moderate degree of eye movement towards these categories. The 'outdoor' category falls slightly behind with a score of 0.58.

Comparing the ASSR dataset with ours, both datasets share similarities in the relative ranks of some categories. For instance, both datasets identify 'person' and 'animal' as highly salient categories, drawing considerable attention. Similarly, categories like 'kitchen' and 'sports' generally receive less attention in both datasets.

However, there are notable differences too. The 'appliance' category, for example, draws significantly more attention in the ASSR dataset compared to ours. 'Electronic' items also seem to draw more attention in our dataset than in the ASSR dataset. These differences may stem from the different modalities of dataset: fixation duration in our dataset versus attention shift from mouse-trajectory-based fixations in the ASSR dataset. This suggests that while some objects inherently draw more attention, the way we measure attention, whether by duration of fixation or by eye movement from mouse-trajectory-based fixations, can also influence the saliency rankings.

4.6.4 Category Complexity

Figure 4-14 shows the average complexity in each of the 12 categories within our proposed dataset. Here, the meaning of complexity is defined by the ratio

of the mean area size to the mean perimeter in instance-level, where the mean area size refers to the average of the total space that each instance occupies within the images and the mean perimeter refers to the average length of the outer boundary of each instance. By dividing the mean area size by the mean perimeter, a simple quantitative measure has been established.

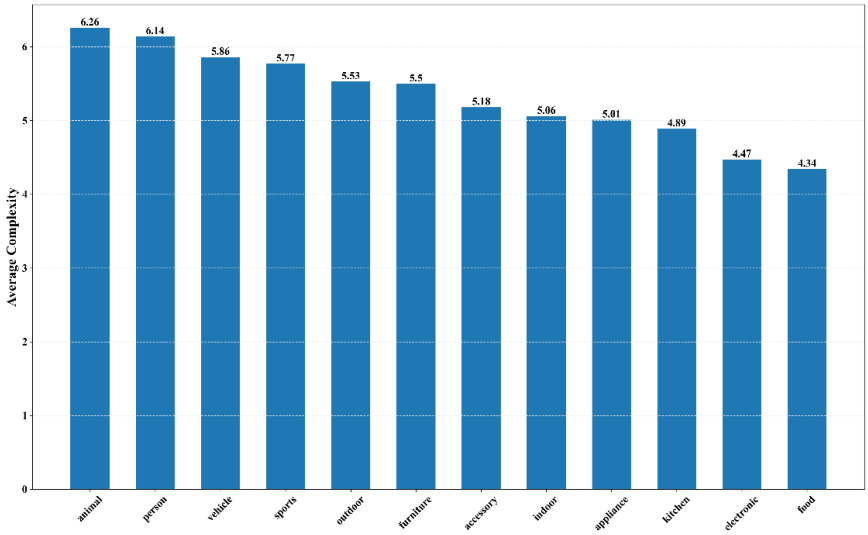


Figure 4-14 Average complexity of each category in our proposed dataset.

Basically, it is theoretically reasonable to posit a correlation between a category's average complexity and its saliency rank. Categories marked by higher complexity may generate a greater degree of attention, leading to a higher saliency rank, as they may make people more curious or require more thinking.

By combining Figure 4-14 and Figure 4-13, it can be observed that categories like 'person' and 'animal' get relatively high complexity scores of 6.14 and 6.26 respectively, which correspondingly demonstrate higher saliency ranks. This could potentially denote that more complicated instances, such as humans and animals, usually attract more substantial attention.

Meanwhile, the 'kitchen' and 'food' categories, with the lowest complexity scores of 4.89 and 4.34 respectively, also present lower saliency ranks. This might suggest that less complex categories are associated with a

reduced degree of attention.

However, it is important to acknowledge the other categories are not following this rule. For instance, the 'electronic' category exhibits a relatively low complexity score of 4.47, yet it commands a high saliency rank. Similarly, the 'accessory' category has a lower complexity score of 5.18 but still achieves a moderate saliency rank. These instances seem to imply that factors beyond complexity also have a significant influence on the relative saliency rank. This observation indicates that complexity alone may not be sufficient to predict the relative saliency rank of an object or category. Other aspects, such as image contrast, instance location, instance size and so on could potentially also play important roles in determining how salient an object appears to the viewer. While complexity in some categories appears to have a correlation with the relative saliency rank, it's important to note that this correlation does not imply a direct reason. The complexity of an object or category is just one of the several reasons that contributes to its saliency. Therefore, considering multiple influencing factors together may provide a more accurate and comprehensive understanding of relative saliency.

4.6.5 Foreground Size

Figure 4-15 shows the statistics on foreground salient objects size ratio on 3 RSR datasets. Here, the foreground salient objects size ratio is defined by the proportion of the total pixels of salient instances in the whole image and the proportion in Y-axis indicates the percentage of images in the whole dataset. The size of the foreground salient objects is categorized into three classes: small, medium, and large. Specifically, small is defined as a foreground size ratio of 5% or less, medium is defined as a foreground size ratio between 5% and 30%, and large is defined as a foreground size ratio of 30% or more. The corresponding data for these categories are presented in

Table 4-2.

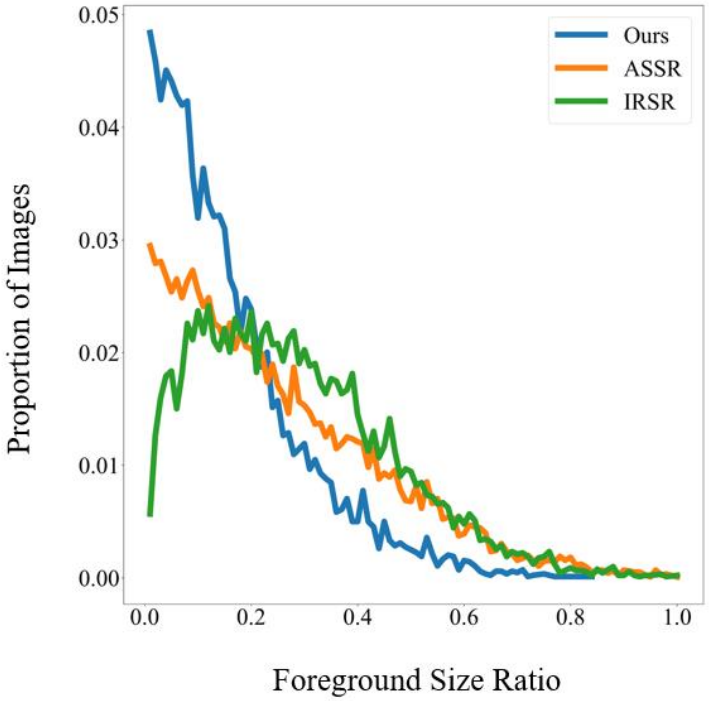


Figure 4-15 Statistics on foreground salient objects size ratio on three RSR dataset.

	Ours	**ASSR**	**IRSR**
Total Images	8389	11500	8988
Large Foreground Images	1292 (15%)	4062 (35%)	3818 (43%)
Medium Foreground Images	5202 (62%)	5855 (51%)	4535 (50%)
Small Foreground Images	1895 (23%)	1583 (14%)	635 (7%)
Maximum Ratio	0.84	1.0	1.0
Minimum Ratio	0.01	0.01	0.01

Table 4-2 Statistics on the foreground size.

As can be observed from the Table 4-2, our dataset, consisting of 8389 images, displays 15% of images with large foreground, 62% with medium foreground, and the remaining 23% with small foreground, approximately subject to a ratio of 1:4:2. The maximum and minimum size ratios are 0.84 and 0.01, respectively. Comparatively, the ASSR dataset, containing 11500 images, exhibits a larger proportion of images with a large foreground (35%),

and a smaller proportion with a medium (51%) and small foreground (14%). The size ratios range from 0.01 to a maximum of 1.0. The IRSR dataset, comprising 8988 images, presents an even higher proportion of images with a large foreground (43%), and similar proportions with a medium (50%) and small foreground (7%) as compared to the ASSR dataset. Like ASSR, the size ratios span from 0.01 to a maximum of 1.0.

A particularly notable feature of our dataset, as revealed in the Figure 4-15 and Table 4-2, is the higher proportion of images with smaller foreground salient objects, accounting for 23% of the total images. In contrast, the corresponding proportions in the ASSR and IRSR datasets are lower, at 14% and 7% respectively. The high proportion of these smaller instances in our dataset implies a higher level of complexity and presents a more challenging scenario for the task of RSR. This is because smaller salient objects can be harder to identify and distinguish.

We argue that our proposed dataset introduces a more challenging task for RSR models, demanding improved sensitivity and precision in recognizing and ranking smaller salient objects spread across an image. Consequently, models trained on our dataset are expected to be more robust and capable of handling a broader range of situations in the real-world applications of RSR.

In addition, another characteristic worth noting is the presence of salient instances with extraordinarily high foreground size ratios in both the ASSR and IRSR datasets. There are 16 images in the ASSR dataset and 6 images in the IRSR dataset where the foreground size ratio exceeds 0.95. The frequency of such images increases even further if we slightly lower the threshold to 0.9 for the foreground size ratio. This phenomenon typically arises due to images that are contaminated by the background, as depicted in as shown in Figure 4-16. In certain cases, the salient object segmentation is not limited to the actual foreground objects but also includes segments of the background. Consequently, this contributes to the corresponding ranking

information, thereby causing negative effects on the overall integrity and reliability of the dataset, which heavily influences the task of correctly identifying and ranking the actual salient objects. This may potentially confound the learning of RSR models, thereby undermining their generalizability and performance in real-world applications.

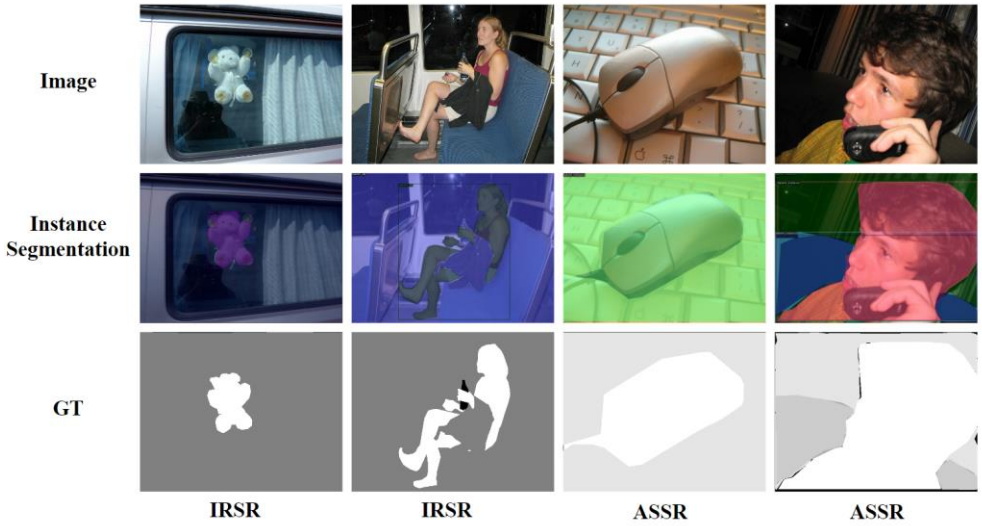


Figure 4-16 Background-contamination examples in IRSR and ASSR.

4.6.6 Instance Size

Similar to Section 4.6.5, the statistics on instance-level size ratio are summarized in Table 4-3. Here, the salient instance size ratio is defined by the proportion of the total pixels of each salient instance in the whole image and the proportion in Y-axis indicates the percentage of instances in the whole dataset. The size of the salient instances is also categorized into three classes: small, medium, and large. Specifically, small is defined as an instance size ratio of 5% or less, medium is defined as an instance size ratio between 5% and 30%, and large is defined as an instance size ratio of 30% or more.

Upon comparison of the three datasets, our proposed dataset exhibits a distinctively high concentration of smaller instances, making up 86.5% of the total instances. This proportion vastly outweighs those in the ASSR and

IRSR datasets, which stand at 70.1% and 55.7% respectively.

Meanwhile, the large instances in our dataset make up a mere 0.2%, significantly less than the 3.7% and 5.1% found in the ASSR and IRSR datasets respectively. The medium instances in our dataset also constitute a lower percentage of 13.3%, compared to 26.2% in the ASSR and 39.2% in the IRSR datasets.

	Ours	**ASSR**	**IRSR**
Total Instances	52173	49445	30176
Large Instances	102 (0.2%)	1852 (3.7%)	1533 (5.1%)
Medium Instances	6924 (13.3%)	12959 (26.2%)	11837 (39.2%)
Small Instances	45147 (86.5%)	34634 (70.1%)	16806 (55.7%)
Maximum Ratio	0.51	0.98	1.0
Minimum Ratio	0.01	0.01	0.01

Table 4-3 Statistics on the instance size.

The dominance of smaller instances in our dataset implies a more challenging situation for the task of RSR. The presence of smaller salient objects necessitates a higher level of precision and sophistication in RSR models, as they need to accurately detect and rank these less pronounced but equally important elements within the image. Therefore, our dataset introduces a more challenging benchmark for the training and evaluation of RSR models, ultimately contributing to their robustness and adaptability in real-world applications.

4.6.7 Instance Location

It is acknowledged that humans have an innate tendency to concentrate their attention on the center of a viewed scene to recognize salient objects. Therefore, it is necessary to evaluate if the datasets are suffering center bias. Following [89][61], the position of salient instances is calculated across the three RSR datasets. Two key metrics are calculated in this process: I_c and I_m, where I_c demonstrates the distance from the instance center point to

the image center point and I_m indicates the distance from the farthest point of the boundary of each instance to the image center point. These generated distances are normalized by dividing each value by half the diagonal length of the image. This normalization process allows for a more meaningful comparison of deviation values across different images in different sizes.

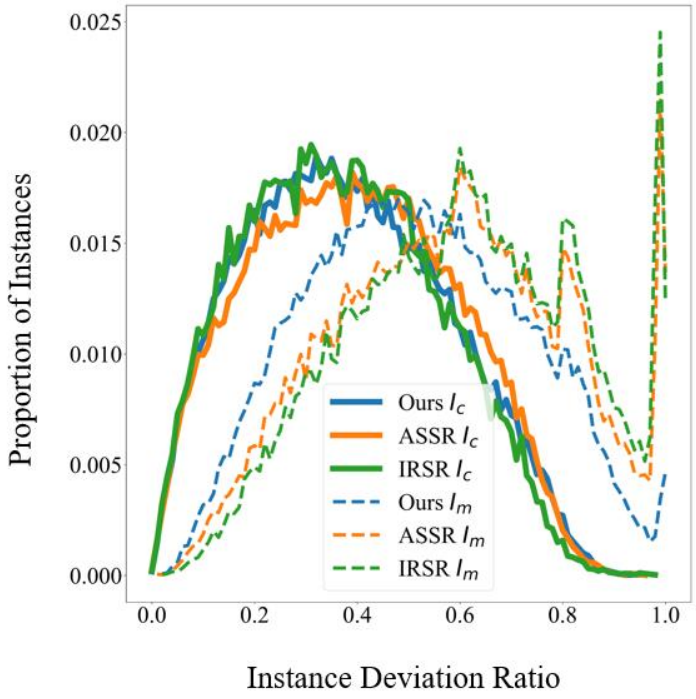


Figure 4-17 Statistics on instance location on three RSR datasets.

Figure 4-17 demonstrates the statistics data based on I_c and I_m of each instance, where y axis indicates the proportion of instances. It can be found from the I_c data that 3 datasets all slightly suffer from the center bias and all 3 datasets generate similar trend in terms of the center deviation of each instance.

The analysis of I_m reveals a more differentiated picture. The ASSR and IRSR datasets show a similar trend, characterized by a big proportion of instances with the farthest boundary points locating at the edges of the visual scenes. This spatial distribution could potentially imply that the salient objects in these datasets are relatively larger or extend towards the boundaries of the images.

On the other hand, our proposed dataset exhibits a different trend. It includes a greater proportion of instances whose farthest boundary points lie within a smaller distance from the image center, indicating a more evenly distributed spatial arrangement of salient objects. Furthermore, a comparison of the I_m and I_c trends in our dataset suggests that the instances are potentially closer together, creating more visually complex and challenging scenes. This characteristic could potentially make our dataset more effective for training models capable of identifying and ranking salient objects in real-world scenarios, where salient objects are often located close together.

4.6.8 Instance Contrast

As suggested by [61], the saliency is related to the global contrast and local contrast in a visual scene. These two elements play important roles in determining the visibility of objects within an image, thereby significantly influencing our perception of saliency.

To quantify these crucial factors, we have calculated both global and local contrasts for each instance in our study. In particular, for global contrast, we first compute the RGB color histograms for both the foreground and background of each instance. The Chi-square (X^2) distance is then used to measure the difference between the foreground and background histograms. This process yields a quantitative measure of the global contrast for each instance.

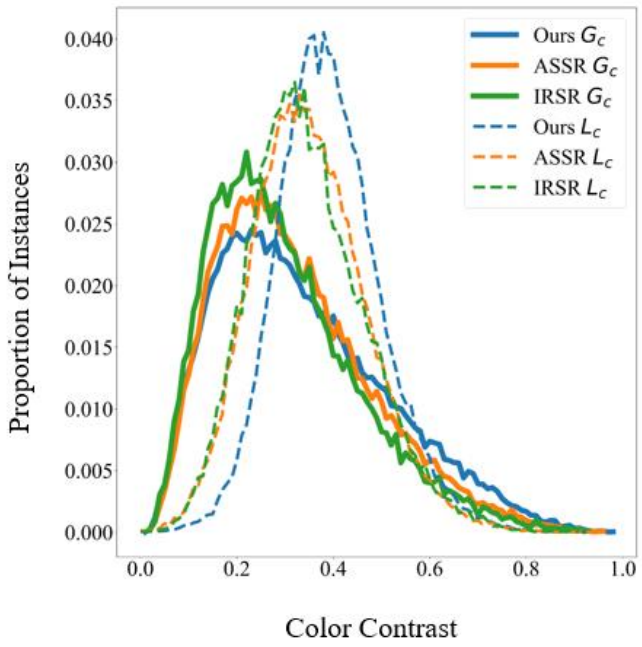


Figure 4-18 Statistics on global contrast and local contrast of each instance on three RSR datasets.

On the other hand, the local contrast is computed by cropping a 5x5 image patch at each boundary point of each salient instance, following the methodology outlined in [61]. Similar to the computation of global contrast, we generate separate RGB color histograms for the foreground and background of these cropped image patches. The Chi-square distance is again utilized to measure the differences between these histograms, thereby providing a numerical representation of the local contrast.

The statistics on global contrast and local contrast in three datasets are shown in Figure 4-18. Interestingly, our proposed dataset exhibits a higher proportion of instances with greater global contrast. This finding suggests that our dataset contains more visually striking and distinguishable objects, which could potentially facilitate more effective and efficient saliency detection.

In terms of local contrast, our dataset also demonstrates a higher frequency of instances with bigger local contrast. Given the manner in which local contrast is calculated, high local contrast implies the presence of well-defined boundaries. Therefore, this observation could be interpreted as an

indication of the superior annotation quality in our dataset.

4.7 Conclusion

In this chapter, we provided a comprehensive overview of our proposed RSR dataset. Our data collection approach integrates the naturally viewing patterns of human observers, offering a closer approximation to real-world perception compared to existing datasets. To the best of our knowledge, this is the first large-scale dataset annotated based on real human fixation patterns for the purpose of RSR.

Our proposed dataset also distinguishes itself from existing datasets by breaking away from the traditional norm of setting an arbitrary limit on the maximum number of salient instances per image. This is a common occurrence in existing datasets, and could potentially restrict the complexity and richness of visual scenes, thus limiting the robustness of saliency detection ranking models. In contrast, all the images in our dataset contain three or more salient instances, providing a more authentic representation of RSR issues and enhancing the diversity and intricacy of the dataset.

The statistical analysis conducted on our dataset, including instance number, categories, size, location, and contrast, highlights the variety and challenge introduced by this new dataset. This level of complexity also offers a more difficult testing ground for new models, which will help drive the development of more robust models.

Chapter 5 Exploration of Transformers for Salient Object Detection

5.1 Introduction

In Chapter 3 we explored the CNN-based architectures to perform MSOD. Our proposed method achieved state-of-the-art performance, exceeding previous models by a large margin. To investigate MSOD task in depth, we created a large-scale instance-level RSR dataset in Chapter 4. Our aim is to develop new and powerful techniques for salient object ranking. Our work in chapter 3 was based on CNNs. More recently, transformer models have moved over from natural language processing, and have produced very impressive results on many visions tasks. Our hypothesis is that a transformer-based approach would perform well on a saliency ranking problem. However, we first explore the use of transformers on the simpler task of SOD. Where instance-level RSR is a combined instance level segmentation and prediction task, SOD can be regarded as a simple binary segmentation problem. One key challenge is that transformers have been seen to produce particularly good results on large datasets, whereas the existing SOD and RSR are comparatively small. In this chapter, we explore the use of a transformer-based network as an alternative to the method outlined in Chapter 3. With limited data, we find that the transformer-based models can achieve state-of-the-art performance in SOD tasks with careful use of a transfer-learning strategy. This offers an interesting alternative, or compliment to, traditional CNNs. The background of transformers is firstly introduced in Section 5.2. Following this, Section 5.3 shows the experiments carried out in evaluating transformers for SOD. We build upon our

experiments here in chapter 5, with a transformer-based approach for salient object ranking.

5.2 Background

Recently, transformers [36] have had a substantial impact on the field of machine translation, specifically in modeling long-range dependencies. Its fundamental mechanism, self-attention, allows transformers to repeatedly stack self-attention layers throughout the architecture. This facilitates the modeling of long-range dependencies at each layer, thereby offering a comprehensive view of data interactions and dependencies. This kind of transformer technique has also attracted the researchers in computer vision community. The Vision Transformer (ViT) [70] was a pioneering work that demonstrate the potential of Transformer techniques to replace standard convolutional operations in deep neural networks, particularly when dealing with large-scale computer vision datasets. Since then, multiple transformer-based models appeared to address different kinds of computer vision problems. Considering transformer's strong ability on feature extraction and long-range dependencies, classic transformers and vision transformers are introduced below.

5.2.1 An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale

The transformer technique is initially introduced in the field of Natural Language Processing (NLP) and has attracted many researchers in the Computer Vision (CV) community. However, applying the Transformers technique to CV does not come without challenges, due to the different types of data structures used in NLP and CV.

In NLP, the data is organized sequentially. On the other hand, the data

in CV is normally spatial, with width and height contributing to the overall structure of an image. These inherent differences pose a unique challenge when attempting to apply transformer techniques, originally designed for sequential data, to two-dimensional image data.

As a result, a crucial step when employing Transformer techniques in CV is to convert visual data into a sequential format.

The work in [70] presented a novel approach to tackle the issue of using the transformer technique with visual data. The researchers divide images into a series of flattened 2D patches, each treated as a token to construct sequential data compatible with the transformer architecture.

This work not only demonstrates the possibility of utilizing the transformer technique in place of traditional convolutional operations but also highlights some of its limitations. When applied to mid-scale datasets, like ImageNet, the transformer's performance was observed to be slightly below that of traditional models such as ResNet. However, an interesting trend is noticed with the increase of dataset scale. As the scale of the dataset grow, the performance of transformer techniques shows a steady improvement. It is eventually able to match, and in some cases, even exceed the performance of current convolutional models. This observation suggests that large-scale datasets might enable transformers to learn important features, such as translation equivariance and locality, similarly to how CNNs operate. This demonstrates the potential of transformer techniques in computer vision tasks, especially when supplemented with large-scale datasets. It also indicates a promising direction for future research and optimization to further enhance the performance of transformer-based models in computer vision applications.

The structure of the Vision Transformer (ViT) begins with an input image of dimensions $H \times W \times C$. This image is initially segmented into N patches, each with dimensions $P \times P \times C$ (or P^2C). These N patches are then combined into a $N \times (P^2C)$ 2D matrix, which is also referred to as 'flattened

patches.' similar to the word vector configuration used in NLP transformers.

It is worth noting that as the patch size P varies, so does P^2C. In order to prevent the architecture from being affected by changes in patch size, this study implements a linear projection. This approach is used to transform different dimensional flattened patches into fixed size vectors, each with a dimension of D. As a result, the initial input of $H \times W \times C$ is transformed into a $N \times D$ 2D matrix, maintaining a consistent input size irrespective of the original patch dimensions.

Transformers have a powerful ability to learn relationships between pairs of features. However, one limitation for transformers is that they cannot learn and understand sequential positional information, as there is no position information in the self-attention mechanism. To overcome this, Position Embeddings are introduced into the architecture. These embeddings are added to different patch embeddings, helping to preserve the positional information within the data structure, thus providing the Transformer with a sense of spatial perception that it wouldn't naturally possess.

Furthermore, an additional learnable class embedding is incorporated as part of the patch embeddings. This class embedding serves an important role in the architecture. Once the encoder has performed information exchange and applied the multi-head self-attention mechanism, the class embedding become a representation of the entire image. This representation not only includes the spatial and feature relationship information within the image but also the classification information.

In this way, the original transformer architecture is adapted to handle computer vision tasks (image classification in this case). By integrating positional and class embeddings, it becomes possible to capture spatial context and categorization information that is critical for image analysis and understanding.

5.2.2 DETR: End-to-End Object Detection with Transformers

This work [71] tackles the problem of object detection. It aims to identify the classification of distinct objects within each image while together generating corresponding bounding boxes that separating these objects. This approach allows for both object classification and localization effectively, meeting the two key objectives of object detection.

Unlike [70], which relies purely on transformer techniques, this study constructs its model in a hybrid manner. It utilizes the strengths of both convolutional operations, which excel at feature extraction, and transformer structure, recognized for the power of modeling long-range dependencies. This allows for a more comprehensive understanding of images, thereby improving the object detection performance. This hybrid approach has inspired many researchers in the computer vision community.

Specifically, an input image is first processed through a ResNet backbone to extract a feature representation. Subsequently, a 1x1 convolutional layer is utilized to reduce the dimensionality of these features from 2048 to a more efficient size of 256. Taking inspiration from ViT [70], a positional encoding is integrated into the feature map. This step is crucial to ensure that spatial information related to the location of features within the image is preserved and can be processed by the transformer architecture.

The sequence length is set to a constant value of 100, considering the maximum of 63 objects annotated in the Microsoft COCO dataset [72]. Consequently, the dimension of the input sequence becomes (b, 100, 256), where b represents the batch size. After the processing of the Transformer encoder, an output sequence of equivalent dimensions (b, 100, 256) is produced.

Regarding the transformer decoder of the DETR model, the object

queries sequence is initially set to a shape of (100, 256). This is then preprocessed into a shape of (b, 100, 256), aligning with the batch size. Cross-attention operation is utilized here. Cross-attention here is a variant of the attention mechanism that connects the encoder and decoder parts of the model. Instead of self-attention, where the model attends to all positions in the same sequence, cross-attention allows the model to focus on different positions in a separate sequence. This is particularly beneficial when there are dependencies or relationships between the elements of two different sequences, which is a common scenario in tasks such as machine translation and question answering. In the context of the DETR model, the decoder incorporates a layer of cross-attention in its structure. The cross-attention mechanism allows the decoder to refer to the output sequence from the encoder, effectively enhancing the understanding of the spatial layout of the image. Each object query in the decoder not only interacts with other queries through self-attention but also attends to all positions in the encoder output via cross-attention.

This process allows the object queries to gather information about the entire spatial layout of the image and integrate it into their predictions. It essentially means that each object query is informed about the positions and features of other objects in the image, allowing for a more robust prediction of object classes and their bounding boxes. The inclusion of the cross-attention layer in the Transformer decoder thus significantly enhances the DETR model's capability to detect multiple objects and their locations in an image.

After the information exchange executed by the transformer decoder, a Feedforward Network (FFN) is utilized to generate the predicted classes and their respective bounding boxes.

In the Microsoft COCO dataset, individual objects are marked with indices ranging from 1 to 91. Thus, considering an additional class for background, the dimension of the class token is configured to be (b, 100, 92).

At the same time, each bounding box is represented by a 4D vector, encoding the bounding box center coordinates and the relative height and width with respect to the image size. As a result, the box token's dimension is set to (b, 100, 4).

The final stage of this process involves the use of the Hungarian algorithm [73], a combinatorial optimization method used to determine the optimal one-to-one match between the model's predictions and the ground truth labels. By utilizing this algorithm, the DETR model can effectively establish a correspondence between predicted and groundtruth objects and followed by supervied loss function to train the object detection task.

In summary, this study presents a solution for object detection challenges using a transformer-based approach, generating comparable results to the optimized Faster RCNN [74] baseline on the COCO dataset. DETR employs a hybrid methodology, incorporating both CNNs and transformers, significantly inspiring researchers in this field and promoting the development of hybrid transformer-based models. This combined approach allows the model to leverage the strengths of both CNNs and transformers, resulting advancements in object detection tasks.

5.2.3 Deformable DETR: Deformable Transformers for End-to-End Object Detection

Deformable DETR is an advanced object detection model that combines the strengths of Transformers and deformable convolutions proposed by Zhu et al., 2021 [97].

Traditional DETR [71] has some limitations. It uses a fixed-size sliding window for self-attention computation, which is not optimal for objects of different sizes and aspect ratios. Moreover, DETR requires a large number of training epochs to converge, which makes it less practical for real-world applications. Deformable DETR addresses these issues by introducing a

modification to the original DETR model.

Deformable Attention Module is proposed in deformable DETR to address the above-mentioned problems. Instead of using all the pixel location to carry out the self-attention mechanism, deformable attention module samples a subset of locations around the reference point for the self-attention mechanism. This operation largely reduces the parameters and the computational complexity of the model.

In deformable attention modules, for each reference point (query feature), the linear layers are utilized to predict the sampling offsets around the reference point. Then, only the offsets-guided points are used to carry out self-attention mechanisms. Other linear layers are also applied based on the reference point to generate the self-attention weights. In this process, only sampled points are utilized.

Deformable DETR further introduces a multi-scale deformable attention mechanism, which allows the model to capture features at different scales. This is particularly useful for detecting small objects and fine-grained details.

In experiments, Deformable DETR has achieved state-of-the-art performance on several benchmark datasets, including COCO and LVIS. It has also demonstrated superior performance in handling objects of different sizes and aspect ratios, as well as in detecting small objects and fine-grained details.

5.2.4 CvT: Introducing Convolutions to Vision Transformers

Despite progress being made in transformer-based models in computer vision, a multitude of unresolved issues continue to persist. Both the pure vision transformer [70] and the hybrid transformer model [71] always require a large volume of data and substantial computational resources to

successfully complete the training process.

In contrast to these transformer models, CNNs tend to extract features in a different manner. CNNs typically identify and process features based on the spatial relationships of neighboring pixels. This is achieved through mechanisms such as local receptive fields (kernals), the use of shared weights, spatial subsampling and so on, the information of which is highly correlated and hierarchical (high-level features and low-level features), therefore making the training of CNN easier, requiring less data.

The work in [75] aims to unify the previously mentioned strengths inherent to CNNs and the positive aspects of transformers. The result of this combination is a model known as the Convolutional Vision Transformer (CvT). This innovative model substantially minimizes the volume of data necessary for training transformer-based models, presenting a significant step forward in addressing the challenges of such models.

One of the key innovations presented in this work is the introduction of the convolutional token embedding module. The purpose of this module is to gradually decrease the size of the input images at each stage of the process, where the feature maps become gradually high-level as they progress through the stages of the CvT. By implementing this method, the length of each sequential token is effectively reduced.

In contrast to previous transformer-based models, which employ positional encoding to retain spatial location information, this research work introduces a unique design called a convolutional projection module. This module, instead of using positional encoding, applies a depth-wise convolutional operation directly to construct the query, key, and value for the multi-head self-attention mechanism. The depth-wise convolutional operation offers a number of benefits. One of the major advantages is its capability to extract spatially neighboring information, i.e. data or features that are in close proximity to each other within the spatial layout of the data, much like how our brains process information in our visual field. Compared

to the traditional positional encoding operation, this method is much more flexible. While positional encoding provides a fixed representation of position, the depth-wise convolutional operation can adapt to different scenarios and input sizes, providing a more dynamic way of handling the positional information. This flexibility can lead to a better understanding and representation of the data, potentially improving the model's performance in various computer vision tasks.

The Convolutional Vision Transformer applies a series of stages in a repetitive manner to progressively extract features from the input data. In a typical CvT-21 model, the three stages contain a varying number of blocks, with stage 1 having 1 block, stage 2 containing 4 blocks, and stage 3 consisting of 16 blocks. These stages, each containing multiple blocks, work in a hierarchical manner, allowing the CvT model to extract more complicated details at each stage. These blocks act as a medium for iterative feature extraction. They break down the input feature maps and extract essential features progressively, starting from simple low-level features and gradually moving towards more complex, high-level representations.

At the final stage, a class token is appended to the sequence. The addition of a class token is a common practice in transformer models used for classification tasks. This class token represents the final output of the model and is used to solve the classification problem. It accumulates information throughout the stages of the model and is ultimately responsible for outputting the prediction. The classification results, derived from the class token, serve as the model's determination of what the input data represents, thereby solving the classification problem that the model was designed to tackle.

In comparison to other transformer-based models, this work proves itself to be more efficient and effective in multiple ways. It requires less data for training, making it more feasible for applications where the data is limited or expensive to gather. Additionally, it demands less computational

power, which is a significant advantage given that computational resources can be costly and are often a limiting factor in model training and deployment. Furthermore, the model contains fewer parameters. This reduction in parameters not only simplifies the model but also helps mitigate overfitting. The model demonstrates state-of-the-art performance across various benchmarks. The efficiencies and effectiveness of this model provide considerable inspiration for the development of future transformer models.

5.2.5 Swin Transformer: Hierarchical Vision Transformer using Shifted Windows

Swin Transformer is a novel vision transformer model that was proposed by Liu et al., 2021 [96]. The name "Swin" stands for "Shifted Window", which reflects the unique design of the model. The Swin Transformer introduces a new mechanism for self-attention computation in Transformers, which is more suitable for image processing tasks.

In ViT model mentioned in Section 5.2.1, the self-attention mechanism computes the attention scores for all pairs of tokens, which can be computationally expensive for large images. The Swin Transformer, on the other hand, divides the input image into non-overlapping windows and computes self-attention within each window, which significantly reduces the computational cost. The Swin Transformer also introduces a hierarchical structure, where the size of the windows is increased, and the number of tokens is reduced in the higher layers of the model. This design is similar to the convolutional layers in CNNs, where the receptive field increases in the higher layers.

For an input image, the Patch Partition operation is firstly applied, which slides a 4x4 non-overlapped window on the input image. In this process, each 4x4 window will flatten the image patch along the channel

dimension. In this case, the RGB 3-dim image patch will be transformed into a 48-dim feature. Then, the Linear Embedding operation is used to transform the 48-dim features to C-dim features.

The Patch Merging operation plays a similar role as the Patch Partition and Linear Embedding, which helps to reduce the width and height of the feature map and at the same time increase the channel numbers. Specifically, the Patch Merging layer concatenates the features of the same location in each 2x2 neighboring patches, followed by a layer normalization operation. Finally, linear layers are used to reduce the channel dimension from $4C$ to $2C$. The whole process is shown in Figure 5-1.

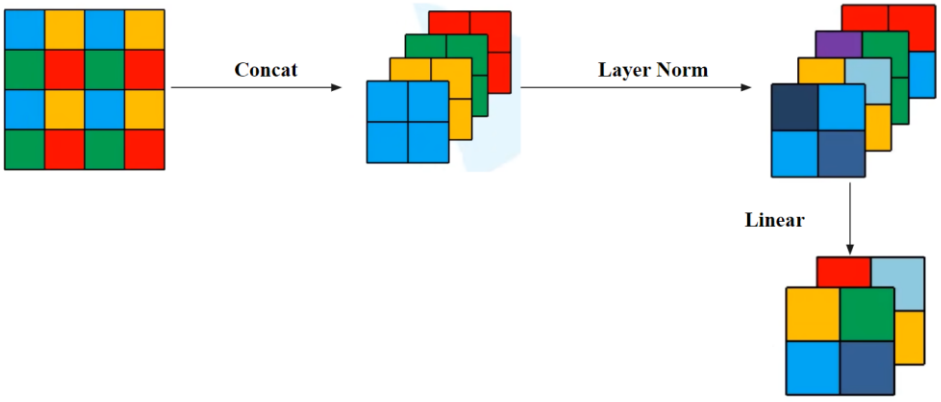


Figure 5-1 The process of Patch Merging.

To reduce the computation, Swin Transformer utilizes W-MSA module to carry out the multi-head self-attention mechanism. Different from the traditional multi-head self-attention that calculate the long-range dependencies across all the pixels in an image, W-MSA only computes the multi-head self-attention within each window, which significantly reduces the computational cost. However, this operation obstructs the communications between different windows.

To solve the problem, in the second transformer block, Shifted Windows Multi-Head Self-Attention (SW-MSA) is proposed. The result generated from layer l only contain the long-range dependencies information within each window. In the following $l+1$ layer, the window

patch is shifted to generate new windows and SW-MSA is calculated based on the new generated windows. In new windows, the calculation of MSA crosses the boundaries of previous windows in layer l, providing information exchange between different windows.

A problem happens in the shifted window is that it will generate more windows and some windows are smaller. To solve this issue, a cyclic shift method is proposed. The new windows generated from SW-MSA is shifted to construct regular patches, and the self-attention operation is utilized in a masked manner to only calculate the MSA in the specific area, which is highly clean and efficient.

The Swin Transformer has achieved state-of-the-art performance on a variety of vision tasks, including image classification, object detection, and semantic segmentation. One of the key advantages of the Swin Transformer is its flexibility and scalability. The model can be easily scaled up or down by adjusting the size of the windows and the number of Transformer layers.

5.2.6 Mask2former: Masked-attention Mask Transformer for Universal Image Segmentation

This work [98] presents a new architecture named Mask2Former, which is capable of addressing any image segmentation task, such as panoptic, instance, or semantic segmentation.

Mask2Former is built upon a simple meta-architecture consisting of a backbone feature extractor, a pixel decoder, and a Transformer decoder.. The authors propose several innovative modules that enable better results and efficient training.

First, to better recognize the small objects, this work proposes an efficient method that utilizes the high-resolution maps from the pixel decoder. Here, the multi-scale deformable attention Transformer [97] is applied in the pixel decoder. Specifically, at each stage, one of the multi-scale feature maps

will be fed into the corresponding layer of the Transformer decoder for the cross-attention based on query feature. This operation helps the model to extract important information from different scale high resolution feature maps, largely improve the performance for detecting small objects.

Second, As discussed in the paper, it should be noticed the importance of context features for image segmentation and the slow convergence of Transformer-based models due to global context in the cross-attention layer. The authors propose masked attention, a variant of cross-attention that only focuses on extracting localized features by limiting cross-attention to the foreground area of each predicted mask, rather than spreading attention across the entire feature map. This operation is carried out based on the attention mechanism between the query feature and the highest resolution feature map in pixel decoder, which will learn to generate a mask for the cross-attention operation in the Transformer decoder layers at each stage.

Third, this work proposes optimization improvements such as switching the order of self and cross-attention, making query features learnable, and removing dropout; all of which improve performance without additional compute.

Finally, Mask2Former saves training memory without affecting the performance by calculating mask loss on a few randomly sampled points. These improvements not only boost the model performance but also make training significantly easier, making universal architectures more accessible to users with limited compute.

One of the key findings in this paper is that the learnable queries can be regarded as region proposals. Region proposals [99], whether they are box-shaped or mask-shaped, represent areas that are probable objects. When learnable queries are supervised by the mask loss, the predictions derived from these queries can act as mask proposals. This finding strongly inspires the future development of Transformer technologies.

Mask2Former achieves state-of-the-art performance, but there are

remaining challenges that Mask2Former faces, particularly in segmenting small objects and fully utilizing multi-scale features. It is recognized by the authors here that better utilization of the feature pyramid and the design of losses specifically for small objects are crucial elements for enhancing its performance.

5.2.7 Summary of Popular Transformer Models

Table 5-1 highlights the summary and comparison of popular transformer techniques, showing how they have evolved to address specific needs in computer vision.

Considering the effectiveness of CvT, efficiency, and the less data required for training, CvT is chosen at the first step to investigate the ability of transformer techniques. The related experiments are introduced in the next section.

Transformer Model	Key Features	Application	Contributions	Potential Limitations
Vision Transformer (ViT)	Divides images into flattened 2D patches; uses positional embeddings.	Image classification at scale.	Pioneering application of transformers in computer vision	requires large-scale datasets for optimal performance.
DETR	Combines CNNs and transformers; positional encoding; Hungarian algorithm for object matching.	Object detection.	Integrates strengths of CNNs and transformers for improved object detection; effective in classification and localization.	Requires significant computational resources; slower convergence compared to traditional models.
Deformable DETR	Deformable attention modules; multi-scale deformable attention.	Advanced object detection.	More efficient self-attention computation; superior in handling objects of different sizes and aspect ratios.	Requires training adjustments for optimal performance.
Convolutional Vision Transformer (CvT)	Convolutional token embedding; convolutional projection for self-attention; class token for classification.	Image classification	Reduces training data volume; unifies strengths of CNNs and transformers; efficient and effective feature extraction.	May struggle with extreme variations in image size and content; further optimization needed for generalization.
Swin Transformer	Shifted window self-attention; hierarchical structure; patch merging and partitioning.	Image classification, object detection, semantic segmentation.	Reduces computational cost; flexible and scalable; state-of-the-art performance across various tasks.	Potential limitations in handling very high-resolution images; complexity increases with scale.
Mask2Former	Multi-scale deformable attention; masked attention for localized features; training optimizations.	Universal image segmentation.	Addresses slow convergence; can be used for various segmentation tasks.	Challenges in segmenting small objects; requires optimization for feature pyramid utilization.

Table 5-1 Summary and comparison of popular transformer techniques including key features, applications, contributions and potential limitations.

5.3 Experiments

The impressive capabilities of transformer-based techniques have served as a strong inspiration for exploring their applications across various fields. Taking into account the efficiency and effectiveness of the Convolutional Vision Transformer (CvT) [75], along with its lesser data requirements for training, we explore the application of CvT in easier task of SOD to investigate the ability of transformer-based techniques.

5.3.1 CvT-21 Backbone

We first construct a simple CvT-21 backbone, with the objective of examining its capacity to learn elements relevant to the SOD problem. The architecture of this CvT-21 is demonstrated in detail in Table 5-2.

The output of the CvT-21, which is a 14x14 feature map, was directly utilized to generate saliency maps of the same dimensions. These generated maps were subsequently trained using corresponding 14x14 ground truth maps.

However, the models built in this manner failed to exhibit any significant learning, yielding disappointing results. This inability to learn could be attributed to the relative insufficiency of the available training dataset for saliency detection. Although CvT-21 models typically require less data for training compared to other models, the available saliency training dataset, which consisted of nearly 10,000 images, was still insufficient for effectively training the transformer-based model. Therefore, using the pretrained CvT-21 model and transfer-learning technique are essential.

CvT-21					
	Output Spatial Size	Output Channel Size	Layer Name	Layer Settings	Number of blocks
Stage1	56 ×56	64	Conv. Embed.	7×7,64,stride 4	
	56 ×56	64	Conv. Proj	3x3,64	1
			MHSA	H1=1,D1=64	
			MLP	R1=4	
stage 2	28 x 28	192	Conv. Embed.	3x3,192,stride 2	
	28 x 28	192	Conv. Proj	3x3,192	4
			MHSA	H2=3,D2=192	
			MLP	R2=4	
stage 3	14x14	384	Conv. Embed.	3x3,384,stride 2	
	14x14	384	Conv. Proj	3x3,384	16
			MHSA	H3=6,D2=384	
			MLP	R3=4	

Table 5-2 Detailed architecture of CvT-21 proposed in [75]. Conv. Embed.: Convolutional Token Embedding. Conv. Proj.: Convolutional Projection. Hi and Di is the number of heads and embedding feature dimension in the ith MHSA module. Ri is the feature dimension expansion ratio in the ith MLP layer.

5.3.2 CvT-21 backbone with Simple Decoder

In order to further explore the ability of transformer-based models to process and learn from smaller datasets, a new experimental setup is designed. This setup takes the form of an encoder-decoder architecture that utilized a pre-trained CvT-21 model as its backbone. In this architecture, the decoder is built using several convolutional layers at each stage. Additionally, it integrates skip-connections from the corresponding sections of the CvT-21 encoder and finally generate the structure similar to a U-Net [3]. The input images and generated predicted maps are set to 224x224.

Luckily, during the training process, the model displays promising indications of learning with the loss observed to gradually decrease.

Meanwhile, various performance evaluation metrics in validation set show a gradual increase. Despite these promising signs, the overall performance of this model is still relatively low (see Table 5-3). However, these results serve as an important indication that transformer-based models can indeed be trained using smaller datasets, particularly when employing a transfer learning strategy. These observations provide a significant inspiration to continue exploring the potential of transformer model. Although the performance levels might currently be sub-optimal, there is a clear indication of learning and potential for improvement, which provides a foundation for continued exploration and refinement in this direction.

Methods	MaxF ↑	MAE ↓
PoolNet (CVPR2019) [22]	0.8048	0.0539
EGNet (ICCV2019) [21]	0.8152	0.0531
MINet (CVPR2020) [25]	0.8097	0.0555
Proposed in Chapter 3	**0.8234**	**0.0530**
Version 1 (Section 5.3.2)	0.666	0.0994

Table 5-3 Performance comparison between other state-of-the-art methods and the proposed architecture here on DUTOMRON dataset, the best result has been marked bold.

5.3.3 CvT-21 backbone with reverse CvT-21 decoder

Here, an architecture makes use of a CvT-21 as encoder and a reverse CvT-21 as the decoder has been explored. It is acknowledged that in a CvT-21 encoder, the token length is gradually decreasing because of the utilized convolutional operations. In the reverse CvT-21 decoder, the bilinear interpolation is applied to gradually upsample the token length (image size), which allows for recovery of the spatial information lost during the encoding process. Moreover, this architecture also incorporates skip connections. These connections are applied from each stage of the encoder to its

corresponding stage in the decoder, allowing for the preservation and utilisation of low-level feature information, which is often lost in deep networks.

Both the input and output sizes are set to 224x224, which provides a balance between computational efficiency and the level of detail in the processed images. Regarding the settings of the transfer learning technique, the models that only the encoder loads pre-trained weights and that both the encoder and decoder load pre-trained weights are investigated. The performance has been shown in Table 5-4. It can be seen both model version 2 and version 3 perform well, even archieve state-of-the-art performance in SOD area, surpassing our proposed method in Chapter 3.

Methods	MaxF ↑	MAE ↓
PoolNet (CVPR2019) [22]	0.8048	0.0539
EGNet (ICCV2019) [21]	0.8152	0.0531
MINet (CVPR2020) [25]	0.8097	0.0555
Proposed in Chapter 3	0.8234	0.0530
Version 2 (Section 5.3.3)	0.8263	0.0517
Version 3 (Section 5.3.3)	**0.8288**	**0.0514**

Table 5-4 Performance comparison between other state-of-the-art methods and the proposed architecture based on DUTOMRON dataset. Here, version 2 indicates the model that only encoder loads pretrained parameters, while version 3 denotes the one that both encoder and decoder load pretrained parameters.

5.3.4 CvT-21 backbone with reverse CvT-21 decoder using channel-wise attention transformer

A transformer block typically encompasses several self-attention operations. However, currently, only spatial self-attention is implemented within these transformer blocks. Therefore, a CvT-based decoder that integrates channel-wise self-attention operations is explored here. The upsampling method in

the decoder employed here continues to be bilinear interpolation. Skip connections are also used from each stage of the encoder to the corresponding place in decoder. Performance can be found in Table 5-5. Note here only the encoder loads pre-trained weights. The model using channel-wise transformer block does not get very strong performance on MaxF but has relatively good MAE.

Methods	MaxF ↑	MAE ↓
PoolNet (CVPR2019) [22]	0.8048	0.0539
EGNet (ICCV2019) [21]	0.8152	0.0531
MINet (CVPR2020) [25]	0.8097	0.0555
Proposed in Chapter 3	**0.8234**	0.0530
Version 4 (Section 5.3.4)	0.8212	**0.0466**

Table 5-5 Performance comparison between other SOD methods and the proposed model (version 4) using channel-wise attention mechanism based on DUTOMRON dataset.

5.3.5 CvT-21 backbone with reverse CvT-21 decoder using channel-wise attention transformer and channel convolution for upsampling

In a CvT-21 architecture, the convolutional embedding layer is used to gradually decrease the token length (image size) and increase the dimensions of channel space, which process the spatial relationships in local space, supporting the spatial long-range dependency information captured in transformer blocks. In comparison, channel-wise transformers can capture the long-range dependencies information in channel space, but no local relationships between each channel can be captured. If regarding a feature map as a cuboid, there might be local neighborhood and local relationship information in channel-space. Therefore, a model is designed here that utilizes convolutional operation on channel-space as the convolutional

embedding layer to gradually process the local relationships between each channel. As shown in Figure 5-2, the mechanism of this operation is very similar to the traditional convolutional operation in spatial space, which can gradually reduce the dimension of channel space and increase the spatial size (image size), therefore making it a tool used for upsampling.

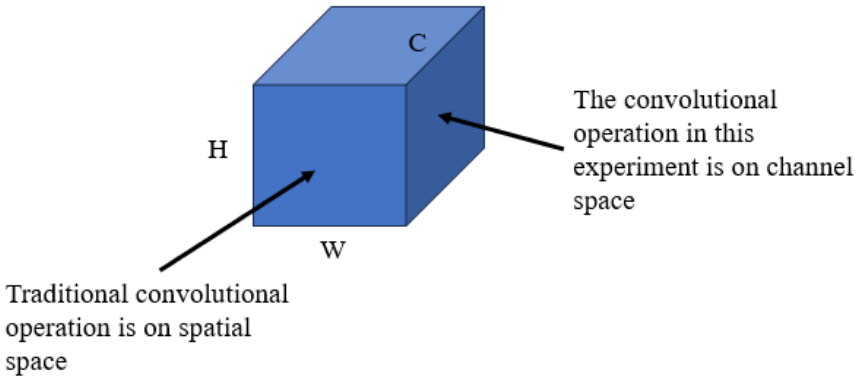


Figure 5-2 Difference between traditional convolutional operation and the proposed convolutional operation on channel space.

The proposed decoder architecture has been shown in Table 5-6. Specifically, the output of the encoder is with dimension $(B, C, H, W) - (B,$ 384,14,14), which is firstly input into an adaptive layer to transfer the channel number to be 576, then, this feature will directly go to the transformer blocks. After stage 1, a convolutional embedding layer will be applied on the channel space of the output of stage 1. Here, the input feature map is of dimension $(B, C, H, W) - (B,$ 576,14,14). The feature is firstly reshaped to be $(B, C, H \times W) - (B,$ 576,196), then transformed to $(B, H \times W, C) - (B,$ 196,576), and finally reshaped based on the square root of channel space to make it $(B, H \times W, c, c) - (B,$ 196,24,24). For an initial test experiment, the channel size should be square rooted. Then, this feature map will be input to the convolutional embedding layer.

The final output is generated after doing a convolutional operation based on the output of this model to generate a single-channel prediction. The performance has been shown in Table 5-8 Version 5. Although the performance is not as well as the model using bilinear operation to upsample

feature maps, it worth exploring as this kind of method does gradually learn something. However, compared to state-of-the-art models, the results generated from this model lack clear edge segmentation.

Adaptive Layer					
	Output Spatial Size	Output Channel Size	Layer Name		Number of blocks
	196 (14x14)	576 (24x24)	Conv.Adaptive	1x1, 576, stride 1	1
Decoder (Channel Transformer)					
	Output Spatial Size	Output Channel Size	Layer Name		Number of blocks
Stage1	196(14x14)	576 (24x24)	Conv. Proj	3x3,576	16
			MHSA	H1=9, D1=576	
			MLP	R1=4	
	784 (28x28)	144 (12x12)	Conv. Embed.	3×3, 784, stride 2	
Stage 2	784 (28x28)	144 (12x12)	Conv. Proj	3x3, 144	4
			MHSA	H2=6, D2=144	
			MLP	R2=4	
	3136 (56x56)	36 (6x6)	Conv. Embed.	3×3, 3136, stride 2	
Stage 3	3136 (56x56)	36 (6x6)	Conv. Proj	3x3, 36	1
			MHSA	H3=3, D3=36	
			MLP	R2=4	
	50176 (224x224)	9 (3x3)	Conv. Embed.	3×3, 50176, stride 2	

Table 5-6 The detailed configuration of the proposed architecture here.

Besides, two different channel-wise transformer mechanisms based on different attention heads separating methods are investigated (Table 5-7). Specifically, **Version 5** is the one using attention heads to separate (divide) the dimension of channel space, while **Version 6** is one using attention heads to separate (divide) the dimension of spatial space. The result has been shown in Table 5-8. As a channel-wise transformer, **Version 6** contains more channel-wise relationships information and thus it achieves better results. This **Version 6** method can be regarded as a way to separate the whole feature map into 4 patches (because of 4 attention heads) spatially and then compute the channel-wise long-range dependencies.

	Input	Query	Attention Heads	Separation (Query, Key)	Attention Map
Version 5	1 196 24 24	1 196 576	9	1 9 196 64	1 9 64 64
Version 6	1 196 24 24	1 196 576	4	1 4 576 49	1 4 576 576

Table 5-7 Two different channel-wise self-attention methods explored here.

Methods	**MaxF ↑**	**MAE ↓**
PoolNet (CVPR2019) [22]	0.8048	0.0539
EGNet (ICCV2019) [21]	0.8152	0.0531
MINet (CVPR2020) [25]	0.8097	0.0555
Proposed in Chapter 3	**0.8234**	**0.0530**
Version 5 (Section 5.3.5)	0.7863	0.451
Version 6 (Section 5.3.5)	0.7934	0.452

Table 5-8 Performance comparison between other state-of-the-art methods and the proposed architecture with two different channel-wise self-attention methods based on DUTOMRON dataset.

5.4 Conclusion

In this chapter, several experiments have been performed to learn and explore the power of transformers for image saliency. We found that the

model with the best performance is the one in which a Convolutional Vision Transformer (CvT-21) is used as both an encoder, and reverse CvT-21 as the decoder. We also found that using the pretrained weights in the decoder in reverse order also improved performance above initializing the decoder from scratch. We hypothesize that these kinds of transformer models, while powerful, are often impractical to be well trained without extremely large datasets. This model achieved state-of-the-art performance in salient object detection on the challenging dataset DUTOMRON. In some configurations, the performance of this transformer approach also surpasses our previous work presented during Chapter 3. In the next chapter, transformer techniques and a transfer learning strategy are combined with graph neural networks and applied to design the architecture for challenging task: relative saliency ranking. As relative saliency ranking is an instance segmentation task and CvT cannot be directly applied for instance segmentation, the mask2former is chosen to be investigated as the detector of relative saliency ranking in the following chapter.

Chapter 6 Instance-Level Relative Saliency Ranking

6.1 Introduction

In previous chapters we have proposed an approach for SOD, new datasets for SOD and RSR, as well as the transformer techniques seen in Chapter 5. In this chapter, we combine a transformer-based detector that carries out instance segmentation with a novel graph reasoning architecture, which learns to rank objects based on their saliency. We train this network on public datasets, and the new proposed dataset presented in chapter 4, where the experimental results demonstrate our proposed method exceeds all previous state-of-the-art approaches for a large margin under three evaluation metrics.

In Section 6.2, the related background technologies of graph reasoning will be described. After this, the research gaps in current instance-level RSR area will be discussed in Section 6.3. The detailed architecture of our proposed method for RSR is introduced in Section 6.4, while Section 6.5 shows the experimental results of our proposed method.

6.2 Background

Saliency ranking task requires not only the detection of different salient objects, but also the relationships between each detected salient objects to generate the relative ranking. Transformer techniques can conduct object detection effectively, while regarding the relationships, graph reasoning methods are good options, which are introduced in the following sections.

6.2.1 Graph Neural Networks

Deep learning has proven to be highly effective in identifying hidden

patterns within Euclidean data [100]. In recent years, inspired by the development of Graph Neural Networks (GNNs) [102], a growing number of applications have represented data as graphs. For instance, facial landmarks can be represented as graphs to better conduct face recognition and face detection [135][136].

In e-commerce systems, a graph-based learning system can leverage the interactions between users and products to generate highly precise recommendations [133]. In the field of chemistry, molecules are depicted as graphs for the discovery of drugs [132].

As shown in Figure 6-1, a graph $G = (V, E)$ is composed of a collection of vertices, denoted as $V \in \{v_i \epsilon R^{1xK}\}$, and edges, denoted as $E \in \{e_{i,j} = e(v_i, v_j) |\, v_i,\, v_j \in V, i \neq j\}$. Here, v_i demonstrates the K attributes of the object within the graph, and $e_{i,j}$ denotes the edge feature that establishes the relationship between vertices v_i and v_j. Each pair of vertices can be linked by a maximum of one undirected edge or two directed edges. A common method to represent such edges is through the adjacency matrix $A \in R^{|v|x|v|}$. In this matrix, all vertices in a graph are arranged such that each vertex corresponds to a specific row and column. Consequently, the existence of each edge $e_{i,j}$ can be indicated by a binary value $A_{i,j} = 1$ if and are v_i and v_j connected, or $A_{i,j} = 0$ if they are not. Notably, the adjacency matrix is always symmetric if all edges are undirected. However, it can be non-symmetric if there are one or more directed edges.

Figure 6-1 Illustration of graph representation [101].

In a typical GNN, the process of learning and prediction involves three

primary steps: aggregation, updating, and looping.

Aggregation is the first step in the process where the GNN collects information from the neighboring nodes of each node in the graph. The purpose of this step is to gather local information around each node using an aggregation function. This function can vary depending on the specific type of GNN, but common functions include sum, mean, or max operations. Consider the graph shown in Figure 6-2. To calculate the updated value of vertex B, the neighborhood vertices C, M and E are all considered to generate neighborhood information, which can be defined as:

$$N_B = f_b(C, M, E) \qquad (6-1)$$

Where N_B indicates the aggregated neighborhood information of B, f_b denotes the aggregation function.

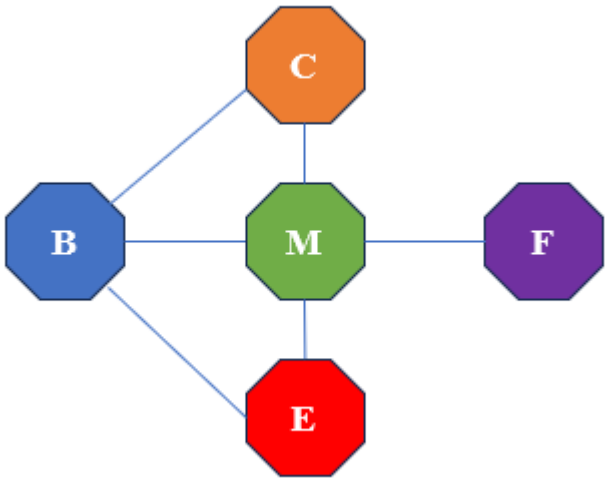


Figure 6-2 An example graph. There are 5 vertices in this graph: B, C, M, E and F.

After the aggregation step, the GNN updates the features of each node based on the aggregated information. This is typically done using a function that takes the current features of a node and the aggregated information as input, and outputs the updated features. This function can be a simple linear transformation followed by a non-linear activation function, or a more complex operation such as a multi-layer neural network. A simple updating

step can be formulated as:

$$B_u = \sigma(WB + \alpha * N_B) \qquad (6-2)$$

Where B_u denotes the updated vertex feature of B, σ is the activation function, W is the learnable weights in the model and α is a weight indicating how much neighborhood information B is going to require.

The aggregation and updating steps are typically performed in a loop for a certain number of iterations. This allows information to propagate through the graph and enables each node to gather information from nodes that are more than one edge away. The number of iterations can be a fixed number, or it can be determined dynamically based on the data.

GNNs have significantly boosted the development of deep learning, particularly for unregular graph-based predictions. An increasing number of research studies are introducing innovative concepts built upon the foundational GNNs for various tasks. Of these, modifications to the aggregation function have gained the most attention.

6.2.2 Graph Attention Networks

Graph Attention (GAT) networks [105] introduce attention mechanism into the aggregation process in traditional GNNs. The input to the Graph Attention Layer $\vec{\mathbf{h}} = \{\vec{h}_1, \vec{h}_2, \cdots, \vec{h}_N\}, \vec{h}_i \in \mathbb{R}^F$ denotes the feature of the vertices, where N is the number of vertices and F is the dimension of the vertex feature. After passing through a Graph Attention Layer, a new feature vector is output, assuming the dimension of this vertex feature is F' (which can be any value), this feature can be represented as $\vec{\mathbf{h}}' = \{\vec{h'}_1, \vec{h'}_2, \ldots, \vec{h'}_N\}, \vec{h'}_i \in \mathbb{R}^{F'}$.

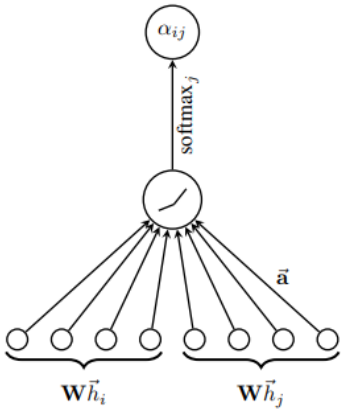


Figure 6-3 The attention mechanism in GAT.

Figure 6-3 demonstrates the architecture of a graph attention layer. In the graph attention layer, a weight matrix $\mathbf{W} \in \mathbb{R}^{F' \times F}$ is first applied to each vertex, and then self-attention is used for each vertex to calculate an attention coefficient. The self-attention mechanism used here can be represented as a:

$$e_{ij} = a\left(\mathbf{W}\overrightarrow{h_i}, \mathbf{W}\overrightarrow{h_j}\right) \qquad (6-3)$$

Where e_{ij} represents the importance of vertex j to vertex i. In theory, the weight of any vertex in the graph to the updated vertex can be calculated. In GAT, in order to simplify the calculation, the vertices are limited to the one-step neighbors of the updated vertex. In addition, the vertex also considers itself as a neighbor vertex. The purpose of using self-attention here is to enhance the vertices feature $\overrightarrow{\mathbf{h}'}$.

There are multiple choices for a, which is parametrized by a learnable weight vector $\vec{a} \in \mathbb{R}^{2F'}$, and then use LeakyReLU, which can be written as:

$$e_{ij} = LeakyReLU\left(\vec{a}^T[\mathbf{W}\overrightarrow{h_i}||\mathbf{W}\overrightarrow{h_j}]\right) \qquad (6-4)$$

Finally, softmax layer is used to normalize the neighbor vertices of the updated vertex:

$$\alpha_{ij} = \text{softmax}j\left(e_{ij}\right) = \frac{\exp\left(e_{ij}\right)}{\sum_{k \in \mathcal{N}i} \exp\left(e_{ik}\right)} \qquad (6-5)$$

Finally, the output feature of updated vertex is obtained by weighting the input vertices features:

$$\vec{\boldsymbol{h}}_i' = \sigma \sum_{j \in \mathcal{N} i} \alpha_{ij} \vec{h}_j \qquad (6-6)$$

Where σ demonstrates the activation function.

This work also demonstrates the effectiveness of GATs through extensive experiments, achieving or matching state-of-the-art results across four established graph benchmarks. This includes tasks such as vertex classification and graph classification, showing the robustness of the proposed approach.

6.2.3 Graph Convolutional Networks

In basic GNNs, the mean operation is commonly used for the aggregation and update operations, however, this can lead to some issues. For instance, considering a graph (see Figure 6-4) including several vertices, to calculate the updated value of vertex B, its unique neighbor vertex M should be firstly found, where M is connecting to many vertices. Normally, in the aggregation and updating period, B will be updated by the mean of B and M. However, it is unfair to B as B is only connected to one vertex M, but M will be influenced by many vertices.

Graph Convolutional Networks (GCN) [104] have been proposed to solve this problem. The layer propagation of multi-layer GCN can be defined as:

$$H^{(l+1)} = \sigma\left(\widetilde{D}^{-\frac{1}{2}}\tilde{A}\widetilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) \qquad (6-7)$$

Where $\tilde{A} = A + I_N$ denotes the adjacency matrix with self-connections, I_N is the identity matrix, $\widetilde{D}$ is degree matrix ($\widetilde{D}_{ii} = \sum_j \tilde{A}_{ij}$) demonstrating the number of neighborhood vertices connected to each vertex, σ is the activation function and $H^{(l)}$ denotes the matrix of

activations in layer l.

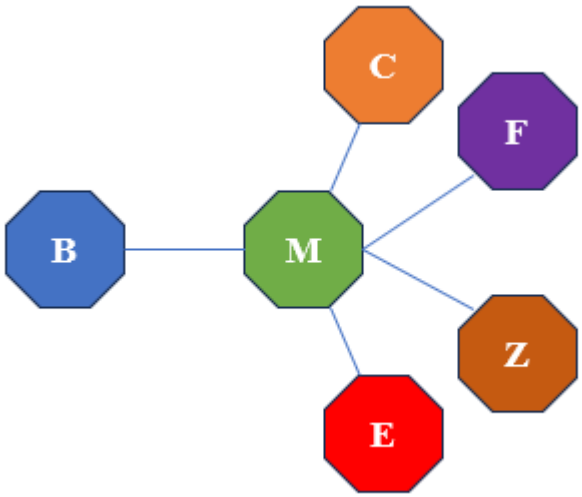


Figure 6-4 Example of the limitations of GNNs.

The aggregation process $\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}$ can be also denoted by:

$$\begin{aligned}\left(\tilde{D}^{-0.5}\tilde{A}\tilde{D}^{-0.5}H\right)_i &= \left(\tilde{D}^{-0.5}\tilde{A}\right)_i\tilde{D}^{-0.5}H \\ &= \left(\sum_k \tilde{D}_{ik}^{-0.5}\tilde{A}_i\right)\tilde{D}^{-0.5}H \\ &= \tilde{D}_{ii}^{-0.5}\sum_j \tilde{A}_{ij}\sum_k \tilde{D}_{jk}^{-0.5}H_j \\ &= \tilde{D}_{ji}^{-0.5}\sum_j \tilde{A}_{ij}\tilde{D}_{jj}^{-0.5}H_j \\ &= \sum_j \frac{1}{\sqrt{\tilde{D}_{ii}\tilde{D}_{jj}}}\tilde{A}_{ij}H_j\end{aligned} \qquad (6-8)$$

Here, it can be found that the aggregation process has been normalized by the degree of the corresponding vertices' neighborhood $\tilde{D}_{ii}\tilde{D}_{jj}$. This normalization strategy ensures that the propagation of information within the graph is not influenced by the degree of the nodes.

6.2.4 Gated Graph ConvNet

The Gated Graph ConvNet (GatedGCN), as described by Bresson and

Laurent [134], is a novel approach based on the idea of GCN, addressing the challenges in the learning on graph-structured data. GatedGCN incorporates key elements such as residual connections, batch normalization, and edge gates to enhance the model's learning capability. These features collectively contribute to a more robust and effective framework.

A distinctive aspect of GatedGCN is the explicit maintenance of edge features alongside node features at each layer. The edge gates in GatedGCN can be perceived as a form of soft attention mechanism, similar to standard sparse attention mechanisms found in other models. This feature is particularly beneficial as it enables the model to know which neighbors in a graph are relevant for a specific learning task, thereby potentially improving performance.

GatedGCN has been shown to outperform other models in several aspects. They are reported to be more accurate than traditional GCNs and faster. The use of gated edges and residual connections plays a pivotal role in these improvements, with residuality being especially crucial in learning multi-layer architectures, leading to a reported 10% gain in performance.

These advancements make GatedGCN a compelling option for tasks involving graph-structured data, such as social networks, brain networks, and other similar domains. The ability to effectively handle variable graph sizes and structures, combined with superior performance, makes GatedGCN as an important contribution to the field of graph neural networks.

6.2.5 Summary of Graph Reasoning Methods

The evolution of graph-based deep learning methodologies highlights a progressive enhancement in handling graph-structured data. GNNs pioneered this domain by enabling the extraction of patterns from graph data. Building upon the foundational GNNs, GATs introduced an attention mechanism to the aggregation process. GATs enhance the model's capability

to weigh the importance of neighboring nodes, allowing for more nuanced information extraction and feature enhancement. GCNs addressed the limitations of basic GNNs, particularly in handling mean aggregation and update operations. GCNs introduced a normalization strategy in the aggregation process, ensuring that the information propagation within the graph is not biased by the nodes' degrees. This innovation provided a more balanced and effective approach to graph representation learning. GatedGCN represents a further advancement, building upon GCN principles. GatedGCNs incorporate residual connections, batch normalization, and edge gates, which collectively enhance learning capabilities.

6.3 Research Gaps

As a pioneering work, RSDNet [81] first proposed the saliency ranking task based on the relative saliency values of different pixels. However, their approach works as a pixel-level relative saliency solution rather than an object-level one that distinguishes between individual object instances. For them to achieve object-level relative saliency rankings, they need to utilize GT instance segmentation maps, making it impractical for real-world applications. ASRNet [78] proposed the instance-level saliency ranking task by deducing the patterns of human attention shifts, illustrating the process of how humans successively choose and divert attention from one object to another. Nevertheless, an object's saliency is predominantly based on gaze duration, rather than the sequence in which objects are observed [110]. IRSR [110] proposed to use graph-reasoning for instance-level RSR. They utilize a person prior in their design, biasing the model towards always predicting people as more salient instances, which may bias against other salient instances. SORNet [112] is the first transformer-based method for RSR. SORNet proposes a position-preserved attention module in the salient object ranking branch to infer spatial positional information. OCOR [113]

introduces a selective object saliency module and an object-context-object relation module to learn the instance-level saliency ranking.

Among these methods, most of them prioritize salient object proposals and exclude the objects with less saliency while building the interactions between objects. Less salient objects that are not involved in interactions, though absent, are still valuable for saliency ranking. Psychological research into visual perception has consistently highlighted the nuanced ways in which humans process and prioritize visual stimuli [117]. While naturally salient objects often capture immediate attention, less salient objects can significantly influence overall scene comprehension and object ranking. For instance, in complex visual scenes, less prominent objects often provide contextual information, aiding in the interpretation and understanding of more salient objects [118]. This interplay between prominent and less-prominent objects mirrors the challenges faced in salient object ranking. In this domain, while the focus has traditionally been on the most salient objects, it is becoming increasingly clear that less salient object proposals should not be overlooked. Their presence and interaction with more prominent objects can be pivotal in determining accurate saliency rankings.

Furthermore, most of the methods set a fixed number of outputs to predict limited saliency rankings, this is because the current datasets also set an arbitrary limit on the salient instances. Setting a fixed number of outputs in saliency prediction models is a simplification that often fails to capture the nuanced ways in which humans prioritize and attend to visual stimuli. Our attention is not limited to a set number of items or regions in our visual field. Instead, it is continuously shifting and adapting based on our goals, experiences, and the context of the environment. For instance, two individuals might look at the same scene but focus on entirely different aspects based on their personal experiences and current emotional state. To truly mimic the human visual system, future saliency prediction methods should consider incorporating more adaptive and flexible mechanisms.

Although current methods have made significant strides in predicting visual saliency rankings, there is still a long way to go before we can truly replicate the intricacies of the human visual system. Therefore, it is crucial to bridge the gap between computational models and human perception.

The task of saliency ranking necessitates not only the identification of distinct salient objects but also an in-depth understanding of the interconnections among these identified objects to establish their relative importance. While transformer-based techniques have proven adept at object detection, they may not adequately address the nuances of relational dynamics between objects. This is where graph reasoning methods are considered in this work, where graph reasoning excels in mapping and interpreting the intricate relationships between objects. This capability stems from its ability to represent objects as nodes and their interrelations as edges within a graph structure. This representation facilitates a more comprehensive analysis of the relational context, allowing for a nuanced understanding of how objects interact within a scene.

6.4 Query as Graph Network

Most of the current salient object ranking methods focus on the building the object interactions based on pre-generated saliency proposals. However, this will ignore the less salient object proposals, which are also valuable for the saliency ranking task. To solve the problem, we propose a Query as Graph Network (QAGNet) built upon the query-based detector [98], ensuring that both prominent and less-prominent objects are considered in the ranking process. Our approach actually supports relative ranking of flexible number of salient instance proposals in a given scene depending on the backbone in use. The overall architecture of one QAGNet layer is shown in Figure 6-5. Given an image, multi-scale features are generated after the backbone [28][96] and pixel decoder [97]. Then, an initial salient instance query $Q_0 \in$

$\mathbb{R}^{N\times D}$ is input into the transformer decoder, where N demonstrates the number of queries and D denotes the feature dimension of salient instance queries. Here, N is set to 100 for ResNet and Swin-B, 200 for Swin-L backbone, while D is set to 256. The learnable salient instance query operates similarly to a region proposal network [74], each of N queries can be regarded as a salient instance proposal $\mathbb{I} \in \mathbb{R}^{1\times D}$ and these N queries include both the prominent and less-prominent instance proposals. Following [98], the multi-scale features from the pixel decoder are fed into 9 transformer decoder layers in sequence, which will generate 9 new salient instance queries, $Q_l^s \in \mathbb{R}^{N\times D}$. These queries represent all potential salient objects throughout different depths in the decoding stage, and at different scales. In our notation, we refer to scales $s \in \{32,64,128\}$ as different stages of the decoder that draw features, via cross-attention, from different resolutions of the feature maps from the pixel decoder. We also denote $l \in \{1,2,3\}$ as the relative position of the layer in the decoding stage. For example, Q_2^{128} denotes the second salient instance query enriched by the feature map of scale 128. Each query vector such as this contains all of the salient instance proposals, for a total of 100 or 200, depending on the backbone network in use. We represent an individual salient instance proposal as $\mathbb{I}_i^{Q_2^{128}}$, corresponding to the ith salient instance proposal belonging to query feature Q_2^{128}. Each of these instance proposals represent one salient object under consideration, and can be used as a feature within our graph structure in order to determine a saliency ranking.

One limitation of [98] is that it commonly struggles to segment smaller objects and does not fully utilize multiscale features. Scale information is an important cue that can be used by humans to judge which instances are more salient. Also, small objects play a significant role in salient ranking tasks, failing to detect such objects will account for false ranking order. We leverage the instance-level salient proposals within the multi-scale query

features as input into a Graph Attention Network (GAT), for better modelling of the saliency ranking problem. We utilise three interconnected graph modules in our work. First, a Single Scale Graph (SSG) combines query features that represent the same scale s, which are combined into representative features for each salient object instance at that scale, serving to enhance the object representation at the same scale. These are then combined within a Multi Scale Graph (MSG), which computes representative features for each salient instance across all scales, aiming to enrich the object representation in a multi-scale view. We call this forward process, from the original query features through to the output of the MSG, the Representative Aggregation (RA) pathway. From this, a Global Representative Graph (GRG) connects all salient instances together, aiming to capture the relationships information between each potential salient objects to learn ranking-aware features. After the GRG process, the query features contain rich ranking-aware information, which is useful both for saliency ranking prediction, but also as feedback to previous query features. We design a Representative Feedback (RF) pathway, which in essence performs the reverse of this hierarchical graph process, which brings the valuable ranking-aware information back to different object feature representations. We iterate this forward-reverse, and for each GRF computed, we can extract a saliency ranking prediction using a ranking head. Note as introduced in [119], the salient instance proposal order here is always in accordance with the initial query Q_0 in the transformer-decoder layers.

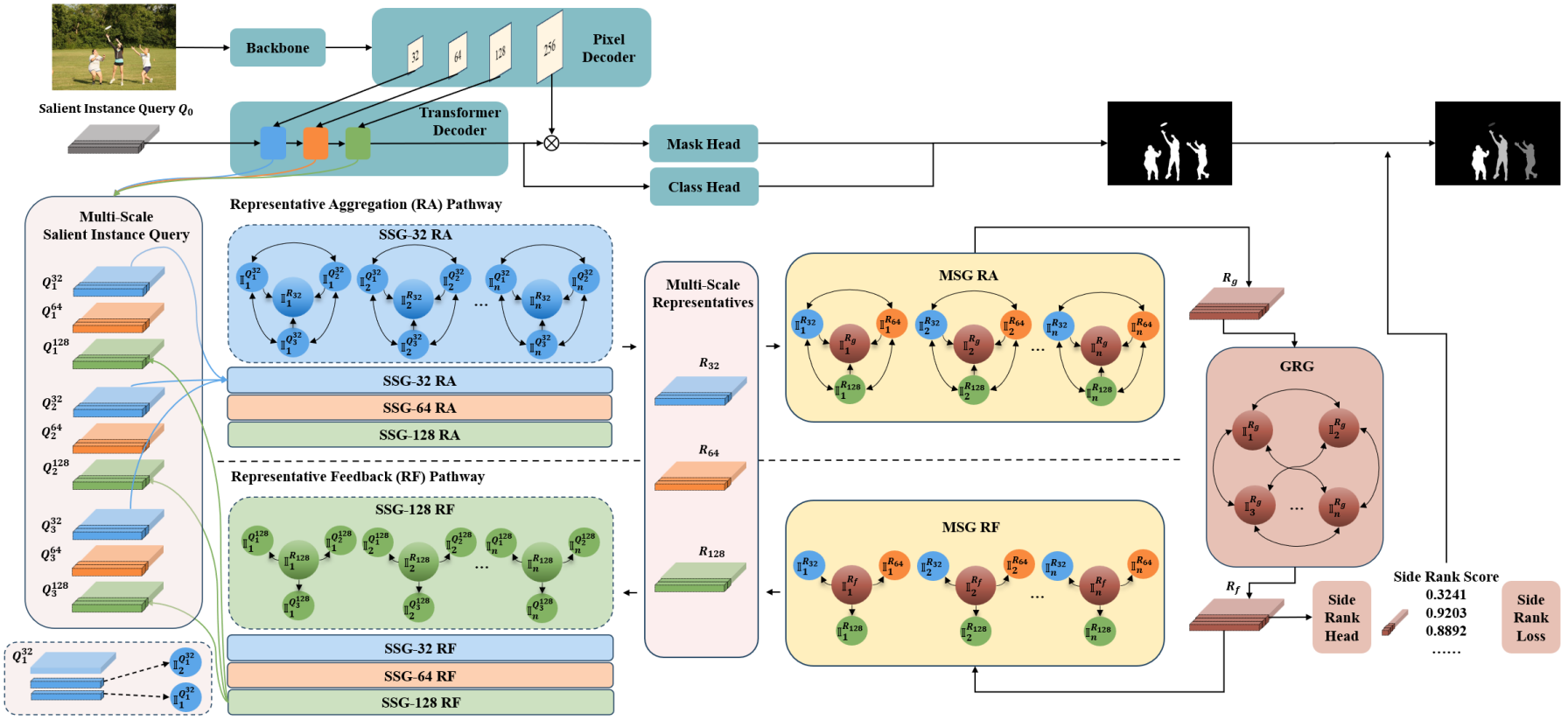


Figure 6-5 The overall architecture of one QAGNet layer. Here, SSG, MSG and GRG demonstrate the Single Scale Graph, Multi Scale Graph and Global Representative Graph.

6.4.1 Representative Aggregation Pathway

We build Presentative Aggregation (RA) pathway to gradually refine the multi-scale salient instance query features. SSG and MSG in the RA process will learn to refine the salient instance proposals at different scales, and generate enriched feature representations respectively, viz., 3 multi-scale query representations $R_s \in \mathbb{R}^{N\times D}$ and global query representation $R_g \in \mathbb{R}^{N\times D}$. As for the representation initialization strategy, we average each 3 identical scale query features to initialize different R_s and average 3 R_s to initialize the global query representation R_g.

6.4.1.1 Single Scale Graph in RA Pathway

Although transformer-based architectures can model long-range dependencies, there might be still information loss across multiple features in cascade-style transformer layers. To fully utilize the multi-scale query features, we firstly build relational SSG to enrich the salient instance queries in same scale. For each scale, we build a Relational SSG (RSSG) $\mathcal{G}^s_{RSSG} = (\mathcal{V}^s_{RSSG}, \mathcal{E}^s_{RSSG})$ to promote the salient instance query features, where $s \in \{32,64,128\}$, $\mathcal{V}^s_{RSSG} = \{\mathbb{I}_i^{Q_l^s}\}_{i=1}^N$ denotes the set of nodes corresponding to the N salient instance proposals in Q_l^s and $\mathcal{E}^s_{RSSG}$ demonstrates the relation edges. In each $\mathcal{G}^s_{RSSG}$, the nodes describing the identical instance in different Q_l^s of same scale will be fully connected, including a self-connection for each instance. In the feature updating process, a salient instance proposal $\mathbb{I}_i^{Q_l^s}$ will be refined by the neighbors from other identical scale salient instance queries representing the same instance proposal. After this, we build directed RA SSGs $\mathcal{G}^s_{RASSG} = (\mathcal{V}^s_{RASSG}, \mathcal{E}^s_{RASSG})$. Here, $\mathcal{V}^s_{RASSG} = \{\mathbb{I}_i^{Q_l^{s\prime}}, \mathbb{I}_i^{R_s}\}_{i=1}^N$ demonstrates the updated nodes in accordance with the N salient instance proposals within Q_l^s and the corresponding instance

representative nodes in R_s, while $\mathcal{E}^s_{N2RISG}$ denotes the directed edges. The updated nodes $\mathbb{I}_i^{Q_i^{s\prime}}$ are pointing to the same salient instance representatives $\mathbb{I}_i^{R_s}$, and finally generating 3 multi-scale representatives R'_{32}, R'_{64} and R'_{128}.

6.4.1.2 Multi Scale Graph in RA Pathway

We model multi-scale instance-level salient proposals as graph here for enhancing the feature representatives of salient instance queries. Similar to SSG, we firstly build Relational MSG (RMSG) as fully-connected graph $\mathcal{G}_{RMSG} = (\mathcal{V}_{RMSG}, \mathcal{E}_{RMSG})$, where $\mathcal{V}_{RMSG} = \{\mathbb{I}_i^{R_s\prime}\}_{i=1}^N$ demonstrates the set of nodes in accordance with the N salient instance proposals with different scales in R'_s and $\mathcal{E}_{RMSG}$ denotes the interactive relation edges. Nodes representing same salient instance proposals in different scales are fully connected including the self-connection. After the information exchange, each node will be leveraged by multi-scale information. We then build RA MSG as directed graph $\mathcal{G}_{RAMSG} = (\mathcal{V}_{RAMSG}, \mathcal{E}_{RAMSG})$, where $\mathcal{V}_{RAMSG} = \{\mathbb{I}_i^{R_s\prime\prime}, \mathbb{I}_i^{R_g}\}_{i=1}^N$ denotes the updated salient instance proposals in R'_s and the ones in global representatives R_g respectively, while $\mathcal{E}_{RAMSG}$ indicates the directed edges. The updated nodes $\mathbb{I}_i^{R_s\prime\prime}$ are directing to the corresponding instance proposal in global representative R_g to forward instance-level multi-scale information.

6.4.2 Global Representative Graph

Now, we get the global representative R_g, each salient instance proposal $\mathbb{I}$ in R_g contain in-depth feature representation from all salient instance queries. We build Global Representative Graph (GRG) as fully connected graph $\mathcal{G}_{GRG} = (\mathcal{V}_{GRG}, \mathcal{E}_{GRG})$ to learn the interactive relationships among all the salient instance proposals, where $\mathcal{V}_{GRG} = \{\mathbb{I}_i^{R_g}\}_{i=1}^N$ demonstrates the N

salient instance proposals in R_g and $\mathcal{E}_{GRG}$ denotes the interactive relation edges. Here, all the salient instance proposals including less salient ones will contribute to the reasoning of ranking. After this, the final feature representative $R_f \in \mathbb{R}^{N\times D}$ is generated. Note in multi-layer QAGNet, short connection is introduced between R_f in current stage and the one in last stage to reduce the rank-aware information loss in the process of information transmission. This operation also serves to highlight the ranking-related feature before the rank head. R_f is then passed to a single linear layer in rank head to predict the rank scores for each salient instance. To enhance the convergence and robustness of model, we also incorporate intermediate ranking loss here.

6.4.3 Representative Feedback Pathway

After GRG, R_f contains rich ranking-aware information gathered from different scale queries. We design Representative Feedback (RF) pathway to transfer the enriched feature back to the various query features though the MSG and SSGs. MSG and SSGs in RF process will feed valuable information back, enhancing the salient instance feature for the next QAGNet layer.

6.4.3.1 Multi Scale Graph in RF Pathway

In RF pathway, we directly build RF MSG as directed graph $\mathcal{G}_{RFMSG} = (\mathcal{V}_{RFMSG}, \mathcal{E}_{RFMSG})$, where $\mathcal{V}_{RFMSG} = \{\mathbb{I}_i^{R_f}, \mathbb{I}_i^{R_s''}\}_{i=1}^{N}$ demonstrates the set of nodes corresponding to the N salient instance proposals in R_f and the hetero-scale instance representatives updated in RA process. $\mathcal{E}_{HSGRFMSG}$ denotes the directed edges. Here, $\mathbb{I}_i^{R_f}$ are directing to $\mathbb{I}_i^{R_s''}$ aiming to backward transmit the global information to multi-scale representatives. This process will update $\mathbb{I}_i^{R_s''}$ and generate new representatives R_s'''. Note

we are not building relational graph here as this will be carried out in the next QAGNet layer.

6.4.3.2 Single Scale Graph in RF Pathway

Finally, we are building directed graphs for different scales in order to pass the global representative information to the same scale salient instance queries. The directed RF SSGs can be denoted by $\mathcal{G}^{s}_{RFSSG} = (\mathcal{V}^{s}_{RFSSG}, \mathcal{E}^{s}_{RFSSG})$, where $\mathcal{V}^{s}_{RFSSG} = \{\mathbb{I}_i^{R_s'''}, \mathbb{I}_i^{Q_l^{s'}}\}_{i=1}^{N}$ demonstrates the updated instance representative nodes in R_s''' and the updated nodes corresponding to the N salient instance proposals within $Q_l^{s'}$ in the RF process, while $\mathcal{E}^{s}_{RFSSG}$ denotes the directed edges. After the information transmission, the new generated 9 salient instance queries are ready for the next QAGNet layer.

6.4.4 Tri-tiered Nested Graph

Building on this, a tri-tiered nested style graph is constructed (see Figure 6-6). Various SSGs serve as subgraphs of MSG to promote the query feature in identical scale. Following this, the MSG acts as a bridge between the SSGs and GRG, carrying out information exchange between query features at difference scales, which can be viewed as the subgraph of GRG. Finally, GRG is designed to model the ranking-aware relationships among all salient instance proposals, facilitating a global information exchange. As can be found that the multi-scale representatives have been updated three times during the whole process of a QAGNet layer, this ensures the multi-scale information are fully captured and utilized. This nested style graph design promotes richer feature aggregation and feedback across multiple scales.

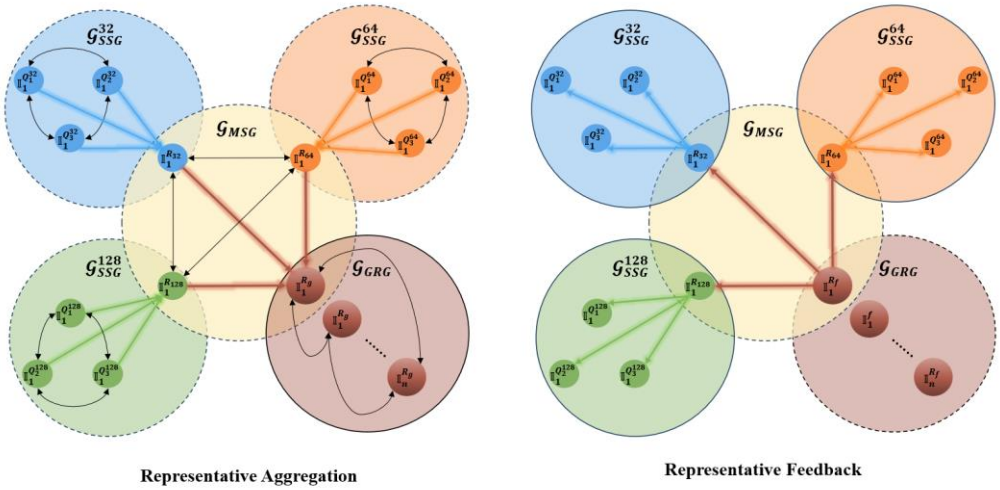


Figure 6-6 Illustration example of the tri-tiered nested style graph.

6.4.5 Multi-layer QAGNet

We experimentally set the total number of layers to two in QAGNet. Specifically, we execute the graph reasoning process, as illustrated in Figure 6-5, twice consecutively. This is then followed by a entire RA process, where the ultimate ranking prediction is derived from R_f in the final GRG. The number of GAT [105] heads in the final GRG are experimentally set to 8 to effectively capturing the diverse features, therefore improving the generalization ability of the proposed method.

Here, the multi-layer design provides a more comprehensive understanding of the data, allowing for more nuanced and precise inferences. Throughout this process, the ranking information of each salient instance is effectively captured. This means that not only does our model gain a more detailed perspective of individual instances, but it also ensures that their relative ranking is duly learned and incorporated. Such a design ensures a harmonious balance between depth and breadth in the learning process, leading to enhanced performance and robustness.

6.5 Experiment

6.5.1 Datasets

To demonstrate the performance of our prosed model, we conduct experiments on two public available saliency ranking datasets ASSR [78], IRSR [110] and our proposed dataset. ASSR provides 7646 training images, 1436 validation images and 2418 test images with at most 5 salient objects per image. IRSR dataset comprises 8,988 images, divided into 6,059 training images and 2,929 test images with at most 8 salient objects. Our proposed dataset contains 6701 images for training and 1688 images for testing. We are not setting an arbitrary maximum number for the ranking of salient objects.

6.5.2 Metrics

We use three widely used metrics in saliency ranking area: Salient Object Ranking (SOR) [81], Segmentation-Aware SOR (SA-SOR) [110] and Mean Absolute Error (MAE).

SOR: The work in [81] proposes several ways to generate rank order. One approach to determine rank order is by averaging the predicted saliency within a specific instance mask. Another proposed method involves assigning rank based on the output from a saliency map, where the saliency degree within an instance is divided by its size, raised to a designed power. Additionally, rank can be determined by considering the peak saliency value within the instance region. This can be formalized as:

$$\text{Rank} = \begin{cases} \text{SOR}_{\text{avg}}\ (\mathcal{S}(\delta)) = & \dfrac{\sum_{i=1}^{\rho_\delta}\ \delta(x_i, y_i)}{\rho_\delta} \\ \text{SOR}_{\text{pow}}\ (\mathcal{S}(\delta); \alpha) = & \dfrac{\sum_{i=1}^{\rho_\delta}\ \delta(x_i, y_i)}{\rho_\delta^{\alpha}} \\ \text{SOR}_{\text{max}}\ (\mathcal{S}(\delta)) = & max\big(\delta(x_i, y_i)\big) \end{cases} \qquad (6-9)$$

Where δ denotes a specific instance from the predicted saliency map $\mathcal{S}$. The value of α is set to 0.3. The term ρ_δ represents the number of pixels within instance δ, while $\delta(x_i, y_i)$ demonstrates the saliency score assigned to the pixel located at coordinates (x_i, y_i).

Here, we follow most of the saliency ranking methods and use $\mathrm{SOR}_{\mathrm{avg}}$ to generate saliency rank order based on generated saliency map for calculating SOR score. Note in instance-level saliency ranking task, the pixel values in an instance are the same. Following this, SOR metric computes the Spearman's rank-order correlation between the prediction and ground truth. However, SOR metric presupposes the predicted instances match the ground truth and only consider the rank orders. High SOR scores can be achieved if the identified salient instances maintain the correct ranking even in the case of missing, incorrect, or low-quality segmentations.

SA-SOR: SA-SOR is proposed in [110], which utilizes the Intersection over Union (IoU) to choose the matched instances and then computes the Spearman's rank-order. SA-SOR also penalizes the missing salient objects and false ranking.

There are a number of issues with SOR metric. SOR metric presumes that the predicted instances align perfectly with those in the GT, focusing solely on rank orders. For instance, even if there are instances that are missed, incorrectly detected, or poorly segmented, high SOR scores can still be achieved as long as the detected salient instances maintain the correct rank order.

For the task of instance-level saliency ranking, it's essential to both segment salient instances and determine their rank order concurrently. Consequently, any evaluation metric used should be sensitive to the quality of segmentation. Siris et al. [78] suggests a method to align GT instance masks with segmented ones. This is done by identifying the segmented instance with the most substantial mask area for each GT mask. This

approach somewhat reduces the issue of the original SOR metric not accounting for segmentation quality. Nonetheless, their simplistic matching technique doesn't ensure a strict one-to-one correspondence, potentially leading to ambiguities. Moreover, they disregard instances that are missed or incorrectly predicted, making their evaluation method not entirely sensitive to segmentation performance.

To solve the problem, the segmentation-aware SOR (SA-SOR) metric is introduced in [110], designed to strictly consider the alignment between segmented salient instances and the GT masks. Specifically, for the segmented instances of a test image, the initial step involves ranking their predicted saliency scores and subsequently assigning these ranks to each instance. For ease of computation and clarity, an ascending rank order is employed for both the predicted and GT ranks. In this metric, higher rank values signify a greater degree of saliency, with 1 being the lowest rank. Following this, the segmented instance masks are matched with the GT masks using an Intersection over Union (IoU) threshold of t, which is set to 0.5. The matching strategy employed is inspired by the strategy used to compute the average precision (AP) metric in instance segmentation tasks. Here, segmented masks are paired with GT masks that meet an overlap criterion, and this pairing is done in descending order based on instance confidence levels. In the end, each GT instance can be paired with a maximum of one segmented instance and vice versa. For two matched masks, their IoU should be at least t or greater.

Following this, the ranks of the matched instances are selected, and the ranks of any missed instances are set to 0. This process results in a predicted rank order. The SA-SOR score is then calculated as the Pearson correlation between this predicted rank order and the GT. Consequently, instances that are segmented redundantly, also known as false positives will disrupt the predicted rank order. Additionally, the missing instances can only be assigned a rank of 0. Both factors will reduce the SA-SOR score, ensuring

that it is highly sensitive to the quality of segmentation. The introduced SA-SOR score promotes both precise salient instance segmentation and accurate ranking orders. For fair comparison, we follow [110] to set the IoU threshold to 0.5 when evaluating using SA-SOR.

MAE: We directly calculate the pixel-level difference between generated ranking saliency maps and ground truth maps. Different salient objects have been assigned different colors, 0 to 255, indicating their RSR in both predicted saliency maps and ground truth maps. As such, MAE here can also reflect the RSR performance.

6.5.3 Implementation Details

The implementation details are introduced below:

Model Settings: The proposed method draws inspiration from the query-based detector Mask2former [98]. As discussed in Chapter 5, transformer techniques require large amount of data to train, we utilize the transfer learning strategy based on the pretrained Mask2former on MS-COCO dataset. Consistent with the configurations of [98], the query number is set to 100 for ResNet [28] and Swin-B [96] backbones, 200 for Swin-L backbone. For our nested-style graph, we incorporate the GAT [105] for both edge calculations and node aggregation. This design has a feature dimension set at 256 and a dropout rate of 0.2. In inference, we determine the final confidence score by multiplying the saliency class confidence with the mask confidence. To ensure precision, only salient instance predictions exceeding a confidence threshold of 0.7 are retained.

Training Settings: The pretrained weights are from the instance segmentation tasks in [98]. We train our model for 30,000 iterations on different datasets. The Adam-W [120] optimizer with weight decay 1×10^{-4} is used to train the network with the learning rate starting from 2.5×10^{-5} and reduced by a factor of 10 at 22,000th and 26,000th iterations.

We resize all the input images to 1024×1024 and do not apply additional pre-processing. We use 4 A6000 GPUs and set the batch size to 4 for Swin-L backbone.

Loss Settings: Following [98], we use binary cross-entropy loss and dice loss [121] for mask and set both weight to 5.0. For rank prediction, we utilize the pair-wise saliency rank loss [110] and set weights to 3.0 for side rank loss and 5.0 for final rank loss. The final loss is a combination of mask loss, saliency classification loss and rank loss.

6.5.4 Comparisons with the State-of-the-Art

We compare our proposed method with 5 state-of-the-art methods: RSDNet [81], ASRNet [78], IRSR [110], SORNet [112] and OCOR [113]. To ensure a fair comparison, we prioritize using the pre-trained models provided by the authors to produce all the results of competing methods. If the pre-trained models on specific datasets are not available, we retrain the model from source code with the recommended settings from the original papers.

For models that output a fixed number of salient objects: RSDNet, ASRNet, SORNet, and OCOR, we adjust this fixed number to match each dataset's maximum instance count. Within RSDNet, we follow [81] to use the stacked representation of the ground-truth with relative setting (illustrated in Section 2.5.1) to regress the saliency values. We then average the predicted saliency values within each instance to get the saliency scores of different instances. As RSDNet cannot predict instance-level masks, we follow [110] to utilize the instance masks from the IRSR to calculate SA-SOR.

For IRSR and our proposed method, both models use a confidence score during inference to select the salient instances. This approach might produce instances surpassing the prescribed limits set on the ASSR and IRSR datasets. Therefore, we follow [110] to present top 5 and top 8 ranked instances in

these datasets as the limited version results. Moreover, results acquired directly post confidence thresholding are reported as the unlimited version.

OCOR employs a low threshold (0.28) in inference to select instances in each rank predictions, which is possible to generate multiple instances for each rank. For fair comparison, we directly report this result as the unlimited version and also report the limited version result by only choosing the highest scoring (bigger than 0.28) instance in different rank predictions. Note that in the OCOR source code, predictions from a higher-ranking instances might be superseded by those from a lower-ranking instance if they pertain to the same instance in the resulting saliency maps.

Currently, in RSR area, researchers typically release pre-trained models and corresponding saliency maps solely for individual datasets. In contrast, we intend to make saliency maps from varied models across different datasets available, paving the way for future investigations.

6.5.4.1 Quantitative Comparisons

In Table 6-1, we demonstrate the quantitative comparison of our method against other saliency ranking approaches. For fair comparison, we evaluate performance across various backbones: ResNet [28], VoVNet [92] and Swin [96]. We also provide information on the number of parameters for each model. It can be found that our proposed method outperforms all other saliency ranking methods by a large margin. The SOR metric does not penalize methods that miss salient instances, so here we focus on the comparison based on SA-SOR.

In the ASSR dataset, compared to the second best model IRSR, our ResNet-based model improves the SA-SOR performance by 8.01%. When it comes to the Swin-based model, our best model surpasses IRSR by 11.0% in terms of SA-SOR. Here, it can be found OCOR gets a very high SOR score, but with a relatively low SA-SOR performance. High SOR scores can

be achieved if the identified salient instances maintain the correct ranking even with the situation of missing, redundant, or low-quality segmentations. This situation is seen on the OCOR result, reflected by its low SA-SOR and our qualitative comparison in the next section. As for the MAE, our proposed method exceeds all other methods dramatically.

On the IRSR dataset, our ResNet-based model improves the SA-SOR score by 8.18% compared to the second best model IRSR, while our Swin-L model pushes this improvement even further, registering a 14.5% growth. Across varying settings, our model maintains its dominant position in the MAE metric.

Regarding our proposed dataset, our proposed method obtains a further performance improvement against competing techniques. Specifically, our ResNet-based model raises the SA-SOR metric by 9.6% against the second-best model IRSR. This gap widens to 11.1% when deploying the Swin-L backbone. Furthermore, it can be found most of the models experience a performance decrease in our proposed dataset, reflecting that our dataset is more challenging and presents greater complexities than ASSR and IRSR.

In our evaluations across all three datasets, it is evident that our ResNet-based QAGNet outperforms other methods in terms of SA-SOR and MAE. As a result, we're introducing the ResNet-based QAGNet as the lightweight edition with only 47.3 million parameters. Additionally, for those seeking more robust performance, we are launching the QAGNet with Swin-Base as the medium-sized variant, housing 110.2 million parameters. Lastly, for maximum performance and comprehensive functionality, we offer the QAGNet based on Swin-Large as our premier model, equipped with 200 queries and a total of 218.8 million parameters.

Method	Backbone	ASSR			IRSR			Our Proposed Dataset			#Para.(M)
		SASOR ↑	SOR ↑	MAE ↓	SASOR ↑	SOR ↑	MAE ↓	SASOR ↑	SOR ↑	MAE ↓	
ResNet and VoVNet											
RSDNet (TPAMI 2019)	ResNet-101	0.6313	0.7758	0.1236	0.4232	0.7096	0.1175	0.4791	0.7239	0.0772	42.7
ASSR (CVPR 2020)	ResNet-101	0.54	0.792	0.101	0.3207	0.6521	0.1098	0.3281	0.5843	0.0624	44.2
IRSR_U (TPAMI 2021)	ResNet-50	0.7051	0.8314	0.0923	0.5647	**0.8143**	0.0953	**0.5585**	0.7487	**0.0465**	128.1
IRSR_L (TPAMI 2021)	ResNet-50	0.709	0.8283	0.0914	0.5648	**0.8141**	0.0953	**0.5585**	0.7487	**0.0465**	
SOR (CVPR 2021)	VoVNet-39	0.6371	0.833	0.0799	0.5171	0.7909	0.0988	0.382	**0.7554**	0.058	119
Our_Model_Res50_U	ResNet-50	**0.7545**	**0.8514**	**0.0619**	**0.611**	0.8108	**0.0845**	**0.6119**	**0.7899**	**0.0437**	47.3
Our_Model_Res50_L	ResNet-50	**0.7658**	**0.8469**	**0.0609**	**0.6107**	0.8106	**0.0845**	**0.6119**	**0.7899**	**0.0437**	
Swin											
OCOR_U (CVPR 2022)	Swin-L	0.6413	**0.8843**	0.0786	0.5183	0.8149	0.1003	0.4392	0.7436	0.0488	401.7
OCOR_L (CVPR 2022)	Swin-L	0.6474	**0.8937**	0.0863	0.5058	0.8184	0.1052	0.4426	0.7462	0.0531	
Our_Model_SwinB_U	Swin-B	0.7741	0.8583	0.0538	0.6252	0.8152	0.0792	**0.6167**	**0.7933**	**0.0409**	110.2
Our_Model_SwinB_L	Swin-B	**0.7809**	0.8529	0.0528	0.6252	0.8151	0.0792	**0.6167**	**0.7933**	**0.0409**	
Our_Model_SwinL_U	Swin-L	0.7793	0.8591	**0.0492**	**0.6466**	**0.8241**	**0.0768**	**0.6206**	**0.7982**	**0.0416**	218.8
Our_Model_SwinL_L	Swin-L	**0.7873**	0.8535	**0.0478**	**0.6468**	**0.824**	**0.0767**	**0.6206**	**0.7982**	**0.0416**	

Table 6-1 Quantitative Comparison with other saliency ranking methods. Different backbones are shown in the 2nd column, e.g., ResNet [28], VoVNet [92] and Swin [96]. We show our proposed method in ResNet-50, Swin-Base and Swin-large. The best two results have been marked as

red and blue. For different methods, the number of parameters is shown in the last column. ↑ indicates the higher the better, while ↓ denotes the lower the better. Note _U and _L indicate the unlimited version and limited version models as illustrated in Section 6.5.4.

6.5.4.2 Qualitative Comparison

In Figure 6-7, we present the qualitative comparison with other saliency ranking methods. As the SA-SOR metric provides a more accurate representation of ranking performance, we select results from the models based on their SA-SOR scores across varying configurations. Note the most salient object is marked as red.

We show the generated results of different models on our proposed dataset from low instance number (left) to high instance number order (right). Multiple challenging images have been shown here, including low-contrast, difficult illumination, small objects and high instance numbers. We can see that our proposed method can generate salient instances with clear boundaries and correct rank orders. Under these challenging conditions, other models can generate results containing superfluous or missing salient instances with incorrect rank order. For example, OCOR sometimes generates results that include incorrect or missing salient instances, but the matched salient objects get right rank order, which leads to a high SOR and a low SA-SOR.

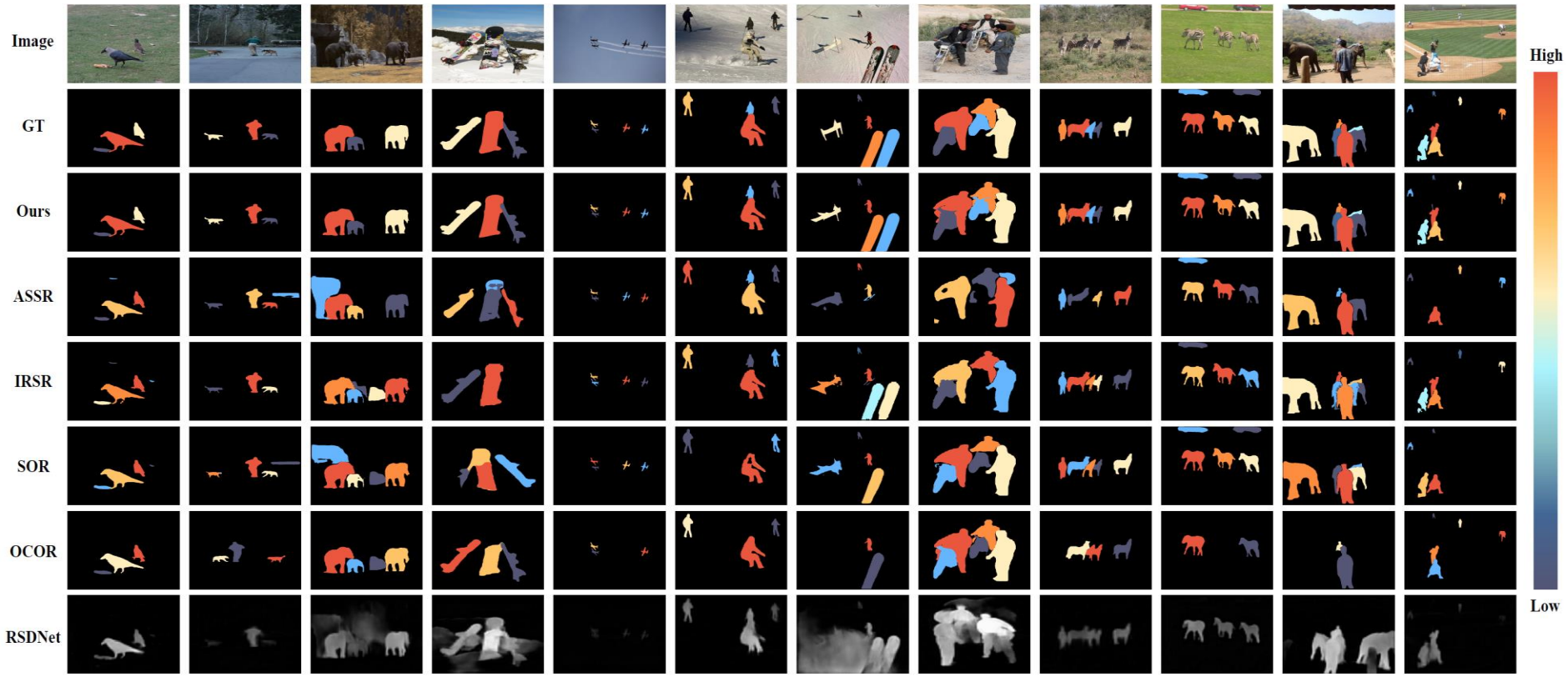


Figure 6-7 Qualitative comparison between our proposed method and other saliency ranking approaches on our proposed dataset.

6.5.5 Ablation Studies

In this section, we investigate the contribution of different model settings on our proposed test set. The experiments here are based on Res-50 backbone.

6.5.5.1 Module Analysis

Setting	Specific Configuration				SASOR	SOR	MAE
I (Baseline)	Last query + Linear layer				0.5623	0.7292	0.0469
II (Baseline)	Average 9 queries + Linear layer				0.5807	0.7381	0.0456
	RA Pathway			RF Pathway			
	SSG	MSG	GRG				
III	✓				0.5837	0.7423	0.0451
IV		✓			0.5944	0.7582	0.0442
V			✓		0.5932	0.7599	0.0445
VI		✓	✓		0.5989	0.7623	0.0441
VII	✓	✓	✓		0.6016	0.7653	0.0442
VIII	✓	✓	✓	✓	**0.6086**	**0.7736**	**0.0439**

Table 6-2 Ablation analysis of different modules in our proposed method.

As shown in Table 6-2, we explore the effectiveness of different modules in our proposed method. Baseline I directly applies a linear layer on the final query feature of [98] to regress the saliency rank scores. In contrast, Baseline II averages all 9 query features before applying the linear layer. It can be found setting II improves the performance, which might because of the information loss in the multi-layer transformer decoder and demonstrates the effectiveness of using all the query features. This observation led us to further investigate the potential benefits of various query features.

Effectiveness of SSG: Setting III only contains a RA pathway with SSG built. The final output is generated based on the mean of all the updated

features from SSG. When comparing settings III with II or VII with VI, the benefits of the SSG become evident. The SSG refines salient instance proposals within the same scale, fostering richer feature representations for subsequent graph reasoning.

Effectiveness of MSG: Comparing setting IV and II or VI and V both clearly demonstrate the effectiveness of MSG. Utilizing MSG increases the performance considerably. Multi-scale representatives are mutually promoted here, helping the model understand saliency features in a multi-scale perspective.

Effectiveness of GRG: The effectiveness of GRG can be observed by comparing the result from setting V and setting II. Specifically, setting V utilize the mean of all query features to construct a fully connected graph. This models instance-level mutual stimuli of human vision systems to learn the multi-relationships among all salient instances.

Effectiveness of MSG & GRG: Solely employing GRG doesn't fully equip the model to grasp multi-instance ranking cues, which can be found by comparing VI and V, e.g., applying both MSG and GRG improve the performance a lot. The MSG plays a significant role in this model, forwarding multi-scale cues to GRG, and therefore promoting the learning process of ranking-aware relationships in multi-scale. Comparing setting V and IV can also potentially verify the importance of using MSG before GRG.

Effectiveness of RA pathway: The effectiveness of RA pathway can be found in setting VII. Using full setting in RA pathway performs even better when compared to using the settings mentioned above, which demonstrates the effectiveness of all the constructed graphs in the representative aggregation process. These modules work together in RA pathway to generate robust feature representatives for ranking prediction.

Effectiveness of RF pathway: Setting VIII integrates the representative feedback pathway to refine query features and followed by a full RA pathway to predict the rankings. This combination forms a complete

QAGNet layer and delivers the best results. This demonstrates the effectiveness of our proposed RF pathway. By feeding the ranking-aware information back, the query feature representatives are gradually polished and passed to the next step of RA pathway. This process helps the model to learn ranking-aware cues in a bi-directional manner.

6.5.5.2 Layer Number and Short Connection Analysis

Setting	Layer Number	Short Connection	SASOR	SOR	MAE
I	1	✓	0.6086	0.7736	0.0439
II	2	✓	**0.6119**	**0.7899**	**0.0437**
III	2		0.6084	0.7803	0.0442
IV	3	✓	0.6089	0.7794	0.044

Table 6-3 Ablation analysis on the layer number and short connection of QAGNet.

In Table 6-3, we delve into how the number of layers and short connection in QAGNet affects its performance. Setting I, which utilizes a single layer QAGNet with short connection applied between the first GRG RA and final GRG, exhibits a performance of 0.6086, 0.7736, and 0.0439 for SASOR, SOR, and MAE metrics respectively. When extended to two layers with short connection, as seen in setting II, there is a noticeable improvement in SASOR and SOR and a slightly better MAE, with values reaching 0.6119, 0.7899 and 0.0437. However, continue adding layers to 3 with short connection (setting IV) does not give performance improvement, which yields a SASOR of 0.6077, SOR of 0.7794 and MAE of 0.044 but still outperforms the single-layer setup considering the metrics that can better reflect saliency ranking performance (SASOR and SOR). By comparing setting II and setting III, the effectiveness of short connection can be observed, which bridges the ranking-aware information between different

stages to highlight the ranking feature before rank head, promoting the model ability to conduct ranking-purpose task. Therefore, we choose 2-layer QAGNet with short connection to continue our experiment.

6.5.5.3 The Number of Salient Instance Queries

Setting	Query Number	SASOR	SOR	MAE
I	50	0.6061	0.7702	0.0435
II	100	0.6119	**0.7899**	0.0437
III	200	**0.6128**	0.7886	**0.0426**

Table 6-4 Ablation study on the number of salient instance queries.

We evaluate the impact of varying the number of salient instance queries on our model's performance in Table 6-4. Setting I, which employs 50 queries, noticeably underperforms compared to setting II with 100 queries. This disparity potentially reflects the significance of considering not only the most salient objects but also those with less saliency. These less salient objects still contain valuable information for instance-level relationship cues, enabling the model to predict more accurate saliency rank. Setting III includes 200 queries, generating the best SASOR score and MAE but lower SOR score. Considering SOR does not pay penalty to the missing salient objects and segmentation quality, we put setting 3 as the premier choice. However, it can be observed that the performance difference between settings II and III remains marginal. Therefore, we employ 100 queries in our lightweight Resnet-50 and Swin-B backbone QAGNet models, while allocating 200 queries to the Swin-L backbone QAGNet for demonstrating the full capability version.

6.5.5.4 Initialization Method for Representative Queries

Setting	Initialization Method	SASOR	SOR	MAE
I	Random	0.6106	0.7872	0.0438
II	Average	**0.6119**	**0.7899**	**0.0437**

Table 6-5 Ablation study on the initialization method for representative queries.

We assess the influence of different initialization methods for representative queries on our model's efficacy in Table 6-5. Specifically, setting I adopt random initialization method for the representative queries while setting II applies the mean method as used in our model. As can be seen, different initialization method has relatively close performance, where setting II is slightly better. This demonstrates the robustness of our proposed method in different initialization method.

6.6 Conclusion

In Chapter 5, we have explored the state-of-the-art transformer technologies, and in this chapter, we have proposed a novel architecture named QAGNet that built upon a strong transformer network (Mask2Former) with graph reasoning methods. Considering training transformer models require a large amount of data and the limited data in RSR area, we utilize the Mask2Former pretrained on MS-COCO dataset as our object detector. In QAGNet, we design a Representative Aggregation pathway and a Representative Feedback pathway that include a Single Scale Graph, Multi Scale Graph and Global Representative Graph. These modules work together to construct a tri-tiered nested style graph that promotes different scale instance-level ranking-ware features, enhancing our model’s ability of correctly detecting multi-scale salient instances and giving accurate rank order. Experiments

shows the effectiveness of different modules in the proposed QAGNet. Our proposed method exceeds all previous state-of-the-art approaches for a large margin under three evaluation metrics. We will release the saliency ranking maps produced by all existing methods on all datasets in this domain. Additionally, we will make the dataset, code, and our pretrained models available.

Chapter 7 Conclusion

7.1 Contributions

In this thesis, we have explored machine learning approaches to SOD and RSR. SOD is usually framed as a binary segmentation task, in which we aim to discover the most salient, or interesting objects in an image. In chapter 3, we introduced a new approach for SOD and tested this on a new curated dataset comprising only complex, multi-object scenes. Saliency object ranking attempts to extend this problem to also separate object instances and rank them in terms of most to least salient. In Chapter 4, we developed a new dataset utilizing human gaze attention to accurately label and rank the importance of salient objects across thousands of images. In Chapter 5 we explored the use of transfer learning for large transformer networks applied to the SOD task, before extending this work on transformers into the full salient object ranking task in Chapter 6. Our results on existing datasets, as well as our new dataset, show leading performance across many metrics compared to existing methods. The primary contributions of this thesis are outlined below:

• A novel MSOD framework is proposed in Chapter 3 that models long-range dependencies in both spatial space and channel space. To the best of our knowledge, this is the first method that explicitly models long-range dependencies in this dual space for standard SOD and MSOD problems. The approach uses non-local guidance and edge refinement modules that work complementarily to enrich feature representations at each stage of the top-down pathway. We curate a new dataset specifically for multi-object saliency problems. Results show that our approach exceeds the performance of 14 state-of-the-art methods across five widely used SOD benchmarks and the proposed multi-object dataset.

• A large-scale instance-level RSR dataset using real human fixations is created in Chapter 4. To the best of our knowledge, this is the first and largest dataset created by real human fixations for RSR. Our data collecting strategy integrates the naturally viewing patterns of human observers, offering a closer approximation to real-world perception compared to other datasets. The focus on challenging multi-object scenes also complements the domain of saliency ranking, and our dataset comprises more objects on average, and a much higher maximum number of objects per scene than existing datasets.

• A novel QAGNet framework is designed and implemented a in Chapter 6 that combines a modern transformer model for instance segmentation, with graph reasoning modules for saliency ranking. The framework draws query features on potential salient objects across different stages and resolutions of the transformer decoder. These are combined to learn the ranking-aware cues within three modules of graph reasoning, from features at the same scale, through features at different scales, and finally features at a global scale. Our results show a substantial jump in performance above other competing methods, including on our newly created ranking dataset. To guide and inspire future research in RSR, we will publish all the saliency ranking maps generated from all the existing methods on all datasets in this domain, as well as the dataset, code, and trained models for our approach.

7.2 Future Work

This thesis delves into the domain of saliency detection, starting from the SOD and then exploring a more complex task of MSOD in Chapter 3. Building upon this foundation, the subsequent chapters investigate the field of RSR and related techniques. RSR is a new task, with many remaining challenges that would be important to explore in the future.

In Chapter 4, a large-scale instance-level RSR dataset using real human

fixations is created. Although we have carefully considered the way we designed the experiment, and utilised the captured data, further improvements to our experimental methodology may be possible in collaboration with experts in human visual saliency from the Psychology field. Such collaboration would offer valuable insights into the understanding of complicated human perception and cognitive processes, enabling the refinement of our experimental design and the enhancement of the dataset's reliability and applicability.

In Chapter 5, popular transformer techniques are investigated, which are instrumental in the detection of objects for the task of RSR. However, transformer models normally require huge amount of data to be well-trained. This requirement poses a challenge in scenarios where data availability is limited such as the saliency ranking area. The exploration of transformer architectures that are efficient with less data is a crucial area for future research. Developing such models would not only enhance the feasibility of deploying transformers in data-scarce environments but also expand their applicability across various domains where extensive datasets are not readily available.

In Chapter 6, a novel architecture using a query-based detector [98] and nested graph neural networks is proposed. This approach has shown strong performance among different the metrics discussed above. In this model, the process initiates with instance segmentation, followed by the implementation of our graph reasoning steps to understand the relationship among different salient instance proposals. The detection and segmentation of salient objects is highly dependent on the accuracy of transformer detector, which might be not always perform perfectly. Further investigation on how to transmit ranking-aware information within our graph reasoning modules back to transformer detectors may improve the accuracy of the detection of salient objects, their segmentation, and their eventual ranking. This could be done by building two pathways to conduct SOD and RSR simultaneously

with information exchange. Regarding the evaluation metrics, three metrics, namely SOR, SA-SOR, and MAE, are commonly utilized in the field of RSR. Among these, MAE is significantly impacted by the quality of segmentation and only marginally reflects the performance of saliency ranking. High SOR scores can be obtained when the salient identified instances maintain accurate ranking even with missing, incorrect, or low-quality segmentations. SA-SOR has been proposed as the solution to these issues. However, while SA-SOR effectively penalizes missing predictions and low-quality segmentations, it does not clearly address the problem of false-positive predictions. These predictions are essentially removed before a rank correlation is computed. There exists a substantial need to explore and develop more appropriate and reliable metrics to solve these limitations to better evaluate the accuracy and reliability of RSR models. Better ranking metrics may also be applicable for techniques that perform object bounding box detection, rather than segmentation, as part of a saliency ranking pipeline.

7.3 Potential Applications

This thesis mainly investigates the area of SOD and RSR. RSR improves upon SOD by differentiating between objects, providing richer representations. RSR can be widely used in numerous downstream tasks, these include object recognition [45], object detection [46][47], image retrieval [48], image captioning [49][50], weakly supervised semantic segmentation[51][52], few-shot learning [131], image cropping [53] and video conversion [126]. Regarding the specific potential applications of RSR, several areas worth for exploration in the future:

Automotive and Transportation Safety: In autonomous driving systems, RSR can be pivotal for identifying and prioritizing critical objects on the road, such as pedestrians, other vehicles, and traffic signs. This aids

in enhancing navigational decisions and overall road safety.

Medical Imaging and Healthcare: RSR can revolutionize diagnostics by pinpointing and ranking areas of interest in medical scans like MRIs or CTs. This application could assist healthcare professionals in quickly identifying critical anomalies, thereby improving the accuracy and efficiency of medical diagnoses.

Retail and E-Commerce: Implementing RSR in online retail platforms could transform user experience by highlighting and ranking products that align with individual preferences. This approach could lead to more personalized shopping experiences and improve the effectiveness of recommendation systems.

Security and Surveillance: In the field of security, especially in crowded or sensitive environments, RSR could enhance surveillance systems. By ranking individuals or objects based on certain criteria, the technology could aid in more effective threat detection and monitoring.

Digital Media and Content Creation: In the area of content creation, including film and advertising, RSR can guide creators in emphasizing elements that capture viewers' attention. This can lead to more engaging and effective visual content.

Robotics and Automation: In robotics, RSR can improve the interaction of robots with their environment, making them more efficient in tasks like object handling, navigation, and interaction with humans or other robots.

Educational Tools and Resources: In educational software and resources, RSR can help create more engaging and interactive learning materials by highlighting key information or concepts in textbooks, instructional videos, or interactive modules.

Bibliography

[1]. Hou, Q., Cheng, M. M., Hu, X., Borji, A., Tu, Z., & Torr, P. H. (2017). Deeply supervised salient object detection with short connections. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3203-3212).

[2]. Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431-3440).

[3]. Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18* (pp. 234-241). Springer International Publishing.

[4]. Zhu, W., Liang, S., Wei, Y., & Sun, J. (2014). Saliency optimization from robust background detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2814-2821).

[5]. Jiang, H., Wang, J., Yuan, Z., Wu, Y., Zheng, N., & Li, S. (2013). Salient object detection: A discriminative regional feature integration approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2083-2090).

[6]. Cheng, M. M., Mitra, N. J., Huang, X., Torr, P. H., & Hu, S. M. (2014). Global contrast based salient region detection. *IEEE transactions on pattern analysis and machine intelligence*, *37*(3), 569-582.

[7]. Wei, Y., Wen, F., Zhu, W., & Sun, J. (2012). Geodesic saliency using background priors. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part III 12* (pp. 29-42). Springer Berlin Heidelberg.

[8]. Yang, C., Zhang, L., Lu, H., Ruan, X., & Yang, M. H. (2013). Saliency detection via graph-based manifold ranking. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3166-3173).

[9]. Li, G., & Yu, Y. (2015). Visual saliency based on multiscale deep features. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5455-5463).

[10]. Wang, L., Lu, H., Ruan, X., & Yang, M. H. (2015). Deep networks for saliency detection via local estimation and global search. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3183-3192).

[11]. Zhao, R., Ouyang, W., Li, H., & Wang, X. (2015). Saliency detection by multi-context deep learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1265-1274).

[12]. Hariharan, B., Arbeláez, P., Girshick, R., & Malik, J. (2015). Hypercolumns for object segmentation and fine-grained localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 447-456).

[13]. Li, G., & Yu, Y. (2016). Deep contrast learning for salient object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 478-487).

[14]. Xie, S., & Tu, Z. (2015). Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision* (pp. 1395-1403).

[15]. Hou, Q., Cheng, M. M., Hu, X., Borji, A., Tu, Z., & Torr, P. H. (2017). Deeply supervised salient object detection with short connections. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3203-3212).

[16]. Hou, Q., Cheng, M. M., Hu, X., Borji, A., Tu, Z., & Torr, P. H. (2017). Deeply supervised salient object detection with short connections. In *Proceedings of the IEEE conference on computer vision*

and pattern recognition (pp. 3203-3212).

[17]. Zhang, L., Dai, J., Lu, H., He, Y., & Wang, G. (2018). A bi-directional message passing model for salient object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1741-1750).

[18]. Zhang, X., Wang, T., Qi, J., Lu, H., & Wang, G. (2018). Progressive attention guided recurrent network for salient object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 714-722).

[19]. Qin, X., Zhang, Z., Huang, C., Gao, C., Dehghan, M., & Jagersand, M. (2019). Basnet: Boundary-aware salient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 7479-7489).

[20]. Feng, M., Lu, H., & Ding, E. (2019). Attentive feedback network for boundary-aware salient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 1623-1632).

[21]. Zhao, J. X., Liu, J. J., Fan, D. P., Cao, Y., Yang, J., & Cheng, M. M. (2019). EGNet: Edge guidance network for salient object detection. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 8779-8788).

[22]. Liu, J. J., Hou, Q., Cheng, M. M., Feng, J., & Jiang, J. (2019). A simple pooling-based design for real-time salient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 3917-3926).

[23]. Wu, Z., Su, L., & Huang, Q. (2019). Stacked cross refinement network for edge-aware salient object detection. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 7264-7273).

[24]. Zhao, X., Pang, Y., Zhang, L., Lu, H., & Zhang, L. (2020). Suppress

and balance: A simple gated network for salient object detection. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16* (pp. 35-51). Springer International Publishing.

[25]. Pang, Y., Zhao, X., Zhang, L., & Lu, H. (2020). Multi-scale interactive network for salient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9413-9422).

[26]. Chen, S., Tan, X., Wang, B., & Hu, X. (2018). Reverse attention for salient object detection. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 234-250).

[27]. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[28]. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

[29]. Yu, F., & Koltun, V. (2015). Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*.

[30]. Zhang, P., Wang, D., Lu, H., Wang, H., & Yin, B. (2017). Learning uncertain convolutional features for accurate saliency detection. In *Proceedings of the IEEE International Conference on computer vision* (pp. 212-221).

[31]. Wang, T., Borji, A., Zhang, L., Zhang, P., & Lu, H. (2017). A stagewise refinement model for detecting salient objects in images. In *Proceedings of the IEEE international conference on computer vision* (pp. 4019-4028).

[32]. Zhao, T., & Wu, X. (2019). Pyramid feature attention network for saliency detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 3085-3094).

[33]. Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, *60*, 91-110.

[34]. Chen, L. C., Papandreou, G., Schroff, F., & Adam, H. (2017). Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*.

[35]. Wang, X., Girshick, R., Gupta, A., & He, K. (2018). Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7794-7803).

[36]. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

[37]. Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., & Lu, H. (2019). Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 3146-3154).

[38]. Borji, A., & Itti, L. (2012). State-of-the-art in visual attention modeling. *IEEE transactions on pattern analysis and machine intelligence*, *35*(1), 185-207.

[39]. Borji, A., Tavakoli, H. R., Sihite, D. N., & Itti, L. (2013). Analysis of scores, datasets, and models in visual saliency prediction. In *Proceedings of the IEEE international conference on computer vision* (pp. 921-928).

[40]. Liu, T., Yuan, Z., Sun, J., Wang, J., Zheng, N., Tang, X., & Shum, H. Y. (2010). Learning to detect a salient object. *IEEE Transactions on Pattern analysis and machine intelligence*, *33*(2), 353-367.

[41]. Achanta, R., Hemami, S., Estrada, F., & Susstrunk, S. (2009, June). Frequency-tuned salient region detection. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 1597-1604). IEEE.

[42]. Liu, F., & Gleicher, M. (2006, July). Region enhanced scale-invariant saliency detection. In *2006 IEEE International Conference on*

Multimedia and Expo (pp. 1477-1480). IEEE.

[43]. Ma, Y. F., & Zhang, H. J. (2003, November). Contrast-based image attention analysis by using fuzzy growing. In *Proceedings of the eleventh ACM international conference on Multimedia* (pp. 374-381).

[44]. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324.

[45]. Rutishauser, U., Walther, D., Koch, C., & Perona, P. (2004, June). Is bottom-up attention useful for object recognition?. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* (Vol. 2, pp. II-II). IEEE.

[46]. Ren, Z., Gao, S., Chia, L. T., & Tsang, I. W. H. (2013). Region-based saliency detection and its application in object recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, *24*(5), 769-779.

[47]. Zhang, D., Meng, D., Zhao, L., & Han, J. (2017). Bridging saliency detection to weakly supervised object detection based on self-paced curriculum learning. *arXiv preprint arXiv:1703.01290*.

[48]. He, J., Feng, J., Liu, X., Cheng, T., Lin, T. H., Chung, H., & Chang, S. F. (2012, June). Mobile product search with bag of hash bits and boundary reranking. In *2012 IEEE conference on computer vision and pattern recognition* (pp. 3005-3012). IEEE.

[49]. Das, A., Agrawal, H., Zitnick, L., Parikh, D., & Batra, D. (2017). Human attention in visual question answering: Do humans and deep networks look at the same regions?. *Computer Vision and Image Understanding*, *163*, 90-100.

[50]. Fang, H., Gupta, S., Iandola, F., Srivastava, R. K., Deng, L., Dollár, P., ... & Zweig, G. (2015). From captions to visual concepts and back. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1473-1482).

[51]. Wang, X., You, S., Li, X., & Ma, H. (2018). Weakly-supervised semantic segmentation by iteratively mining common object features. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1354-1362).

[52]. Wei, Y., Liang, X., Chen, Y., Shen, X., Cheng, M. M., Feng, J., ... & Yan, S. (2016). Stc: A simple to complex framework for weakly-supervised semantic segmentation. *IEEE transactions on pattern analysis and machine intelligence*, *39*(11), 2314-2320.

[53]. Wang, W., Shen, J., & Ling, H. (2018). A deep network solution for attention and aesthetics aware photo cropping. *IEEE transactions on pattern analysis and machine intelligence*, *41*(7), 1531-1544.

[54]. Peng, C., Zhang, X., Yu, G., Luo, G., & Sun, J. (2017). Large kernel matters--improve semantic segmentation by global convolutional network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4353-4361).

[55]. Li, A., Qi, J., & Lu, H. (2020). Multi-attention guided feature fusion network for salient object detection. *Neurocomputing*, *411*, 416-427.

[56]. Sun, F., Li, W., & Guan, Y. (2019). Self-attention recurrent network for saliency detection. *Multimedia Tools and Applications*, *78*, 30793-30807.

[57]. Zhou, Z., Wang, Z., Lu, H., Wang, S., & Sun, M. (2020, April). Multi-type self-attention guided degraded saliency detection. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 34, No. 07, pp. 13082-13089).

[58]. Liu, N., Zhang, N., & Han, J. (2020). Learning selective self-mutual attention for RGB-D saliency detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 13756-13765).

[59]. Wang, L., Lu, H., Wang, Y., Feng, M., Wang, D., Yin, B., & Ruan, X. (2017). Learning to detect salient objects with image-level

supervision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 136-145).

[60]. Yan, Q., Xu, L., Shi, J., & Jia, J. (2013). Hierarchical saliency detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1155-1162).

[61]. Li, Y., Hou, X., Koch, C., Rehg, J. M., & Yuille, A. L. (2014). The secrets of salient object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 280-287).

[62]. Movahedi, Vida, and James H. Elder. "Design and perceptual validation of performance measures for salient object segmentation." *2010 IEEE computer society conference on computer vision and pattern recognition-workshops*. IEEE, 2010.

[63]. Fan, D. P., Cheng, M. M., Liu, Y., Li, T., & Borji, A. (2017). Structure-measure: A new way to evaluate foreground maps. In *Proceedings of the IEEE international conference on computer vision* (pp. 4548-4557).

[64]. Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems* 25 (2012).

[65]. Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[66]. Liu, N., Han, J., & Yang, M. H. (2018). Picanet: Learning pixel-wise contextual attention for saliency detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3089-3098).

[67]. Wu, Z., Su, L., & Huang, Q. (2019). Cascaded partial decoder for fast and accurate salient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 3907-3916).

[68]. Zhao, X., Pang, Y., Zhang, L., Lu, H., & Zhang, L. (2020). Suppress and balance: A simple gated network for salient object detection. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16* (pp. 35-51). Springer International Publishing.

[69]. Yu, S., Zhang, B., Xiao, J., & Lim, E. G. (2021, May). Structure-consistent weakly supervised salient object detection with local saliency coherence. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 35, No. 4, pp. 3234-3242).

[70]. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

[71]. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020, August). End-to-end object detection with transformers. In *European conference on computer vision* (pp. 213-229). Cham: Springer International Publishing.

[72]. Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13* (pp. 740-755). Springer International Publishing.

[73]. Kuhn, H. W. (2005). The Hungarian method for the assignment problem. *Naval Research Logistics (NRL)*, *52*(1), 7-21.

[74]. Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, *28*.

[75]. Wu, H., Xiao, B., Codella, N., Liu, M., Dai, X., Yuan, L., & Zhang, L. (2021). Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF international conference on computer*

vision (pp. 22-31).

[76]. Islam, M. A., Kalash, M., & Bruce, N. D. (2018). Revisiting salient object detection: Simultaneous detection, ranking, and subitizing of multiple salient objects. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7142-7150).

[77]. Li, Y., Hou, X., Koch, C., Rehg, J. M., & Yuille, A. L. (2014). The secrets of salient object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 280-287).

[78]. Siris, A., Jiao, J., Tam, G. K., Xie, X., & Lau, R. W. (2020). Inferring attention shift ranks of objects for image saliency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 12133-12143).

[79]. Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13* (pp. 740-755). Springer International Publishing.

[80]. Jiang, M., Huang, S., Duan, J., & Zhao, Q. (2015). Salicon: Saliency in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1072-1080).

[81]. Kalash, M., Islam, M. A., & Bruce, N. D. (2019). Relative saliency and ranking: Models, metrics, data and benchmarks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(1), 204-219.

[82]. Islam, M. A., Kalash, M., & Bruce, N. D. (2018). Revisiting salient object detection: Simultaneous detection, ranking, and subitizing of multiple salient objects. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7142-7150).

[83]. Van der Lans, R., Wedel, M., & Pieters, R. (2011). Defining eye-fixation sequences across individuals and tasks: the Binocular-Individual Threshold (BIT) algorithm. *Behavior research methods*, *43*,

239-257.

[84]. Rayner, K. (1998). Eye movements in reading and information processing: 20 years of research. *Psychological bulletin*, *124*(3), 372.

[85]. Salvucci, D. D., & Goldberg, J. H. (2000, November). Identifying fixations and saccades in eye-tracking protocols. In *Proceedings of the 2000 symposium on Eye tracking research & applications* (pp. 71-78).

[86]. Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996, August). A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd* (Vol. 96, No. 34, pp. 226-231).

[87]. MacQueen, J. (1967, June). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* (Vol. 1, No. 14, pp. 281-297).

[88]. He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 2961-2969).

[89]. Fan, D. P., Cheng, M. M., Liu, J. J., Gao, S. H., Hou, Q., & Borji, A. (2018). Salient objects in clutter: Bringing salient object detection to the foreground. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 186-202).

[90]. Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, *28*.

[91]. Hafiz, A. M., & Bhat, G. M. (2020). A survey on instance segmentation: state of the art. *International journal of multimedia information retrieval*, *9*(3), 171-189.

[92]. Lee, Y., & Park, J. (2020). Centermask: Real-time anchor-free instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 13906-13915).

[93]. Tian, Z., Shen, C., Chen, H., & He, T. (2019). Fcos: Fully

convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 9627-9636).

[94]. Wang, X., Kong, T., Shen, C., Jiang, Y., & Li, L. (2020). Solo: Segmenting objects by locations. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16* (pp. 649-665). Springer International Publishing.

[95]. Chen, H., Sun, K., Tian, Z., Shen, C., Huang, Y., & Yan, Y. (2020). Blendmask: Top-down meets bottom-up for instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8573-8581).

[96]. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., ... & Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 10012-10022).

[97]. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., & Dai, J. (2020). Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*.

[98]. Cheng, B., Misra, I., Schwing, A. G., Kirillov, A., & Girdhar, R. (2022). Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 1290-1299).

[99]. Uijlings, J. R., Van De Sande, K. E., Gevers, T., & Smeulders, A. W. (2013). Selective search for object recognition. *International journal of computer vision*, *104*, 154-171.

[100]. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Philip, S. Y. (2020). A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, *32*(1), 4-24.

[101]. Song, S., Song, Y., Luo, C., Song, Z., Kuzucu, S., Jia, X., ... &

Gunes, H. (2022). Gratis: Deep learning graph representation with task-specific topology and multi-dimensional edge features. *arXiv preprint arXiv:2211.12482*.

[102]. Gori, M., Monfardini, G., & Scarselli, F. (2005, July). A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.* (Vol. 2, pp. 729-734). IEEE.

[103]. Deng, B., French, A. P., & Pound, M. P. (2023). Addressing multiple salient object detection via dual-space long-range dependencies. *Computer Vision and Image Understanding*, 103776.

[104]. Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

[105]. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. *arXiv preprint arXiv:1710.10903*.

[106]. Amirul Islam, M., Rochan, M., Bruce, N. D., & Wang, Y. (2017). Gated feedback refinement network for dense image labeling. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3751-3759).

[107]. Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2117-2125).

[108]. Judd, T., Ehinger, K., Durand, F., & Torralba, A. (2009, September). Learning to predict where humans look. In *2009 IEEE 12th international conference on computer vision* (pp. 2106-2113). IEEE.

[109]. Yang, C., Zhang, L., Lu, H., Ruan, X., & Yang, M. H. (2013). Saliency detection via graph-based manifold ranking. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3166-3173).

[110]. Liu, N., Li, L., Zhao, W., Han, J., & Shao, L. (2021). Instance-level relative saliency ranking with graph reasoning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(11), 8321-8337.

[111]. Chen, W., Fu, Z., Yang, D., & Deng, J. (2016). Single-image depth perception in the wild. *Advances in neural information processing systems*, *29*.

[112]. Fang, H., Zhang, D., Zhang, Y., Chen, M., Li, J., Hu, Y., ... & He, X. (2021). Salient object ranking with position-preserved attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 16331-16341).

[113]. Tian, X., Xu, K., Yang, X., Du, L., Yin, B., & Lau, R. W. (2022). Bi-directional object-context prioritization learning for saliency ranking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 5882-5891).

[114]. Fang, Y., Yang, S., Wang, X., Li, Y., Fang, C., Shan, Y., ... & Liu, W. (2021). Instances as queries. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 6910-6919).

[115]. Li, P., Xie, J., Wang, Q., & Zuo, W. (2017). Is second-order information helpful for large-scale visual recognition?. In *Proceedings of the IEEE international conference on computer vision* (pp. 2070-2078).

[116]. Wang, Q., Xie, J., Zuo, W., Zhang, L., & Li, P. (2020). Deep cnns meet global covariance pooling: Better representation and generalization. *IEEE transactions on pattern analysis and machine intelligence*, *43*(8), 2582-2597.

[117]. Balcetis, E., & Dunning, D. (2006). See what you want to see: motivational influences on visual perception. *Journal of personality and social psychology*, *91*(4), 612.

[118]. Healey, C., & Enns, J. (2011). Attention and visual memory in visualization and computer graphics. *IEEE transactions on*

visualization and computer graphics, *18*(7), 1170-1188.

[119]. Wang, Y., Xu, Z., Wang, X., Shen, C., Cheng, B., Shen, H., & Xia, H. (2021). End-to-end video instance segmentation with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8741-8750).

[120]. Loshchilov, I., & Hutter, F. (2017). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

[121]. Milletari, F., Navab, N., & Ahmadi, S. A. (2016, October). V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)* (pp. 565-571). Ieee.

[122]. Kümmerer, M., & Bethge, M. (2021). State-of-the-art in human scanpath prediction. *arXiv preprint arXiv:2102.12239*.

[123]. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., ... & Bengio, Y. (2015, June). Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning* (pp. 2048-2057). PMLR.

[124]. Yao, T., Pan, Y., Li, Y., & Mei, T. (2018). Exploring visual relationship for image captioning. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 684-699).

[125]. Chen, J., Bai, G., Liang, S., & Li, Z. (2016). Automatic image cropping: A computational complexity study. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 507-515).

[126]. Zhu, T., Zhang, D., Hu, Y., Wang, T., Jiang, X., Zhu, J., & Li, J. (2021). Horizontal-to-vertical video conversion. *IEEE Transactions on Multimedia*, *24*, 3036-3048.

[127]. Solera, F., & Cucchiara, R. (2019). Predicting the Driver's Focus of Attention: The DR (eye) VE Project. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, *41*(7).

[128]. Matin, E. (1974). Saccadic suppression: a review and an analysis. *Psychological bulletin*, *81*(12), 899.

[129]. Schütz, A. C., Braun, D. I., & Gegenfurtner, K. R. (2011). Eye movements and perception: A selective review. *Journal of vision*, *11*(5), 9-9.

[130]. Danion, F. R., & Flanagan, J. R. (2018). Different gaze strategies during eye versus hand tracking of a moving target. *Scientific reports*, *8*(1), 10059.

[131]. Song, H., Deng, B., Pound, M., Özcan, E., & Triguero, I. (2022). A fusion spatial attention approach for few-shot learning. *Information Fusion*, *81*, 187-202.

[132]. Jiang, D., Wu, Z., Hsieh, C. Y., Chen, G., Liao, B., Wang, Z., ... & Hou, T. (2021). Could graph neural networks learn better molecular representation for drug discovery? A comparison study of descriptor-based and graph-based models. *Journal of cheminformatics*, *13*(1), 1-23.

[133]. Wu, S., Sun, F., Zhang, W., Xie, X., & Cui, B. (2022). Graph neural networks in recommender systems: a survey. *ACM Computing Surveys*, *55*(5), 1-37.

[134]. Bresson, X., & Laurent, T. (2017). Residual gated graph convnets. arXiv preprint arXiv:1711.07553.

[135]. Sun, Z., & Tzimiropoulos, G. (2022). Part-based face recognition with vision transformers. *arXiv preprint arXiv:2212.00057*.

[136]. Sun, Z., Feng, C., Patras, I., & Tzimiropoulos, G. (2024). LAFS: Landmark-based Facial Self-supervised Learning for Face Recognition. *arXiv preprint arXiv:2403.08161*.