Imperial College London
Department of Electrical and Electronic Engineering

Effective Deep Leaning Methodologies for Salient Object Detection

Guangyu Ren

Supervised by Dr. Tania Stathaki

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Computing of the University of London and
the Diploma of Imperial College, February 2023

Abstract

Salient object detection has achieved great improvement by using deep neural networks. Diverse challenging problems arise from this computer vision task and attract more attention from researchers. This thesis investigates the issues in specific tasks and aims at proposing effective methodologies to tackle the problems and improve detection performance.

More specifically, existing network architectures may cause dilution problems in high-level semantic information during up-sample operations in the top-down pathway of the Feature Pyramid Network (FPN). In order to overcome this limitation, we propose a novel pyramid self-attention module (PSAM) and the adoption of an independent feature-complementing strategy. In addition, a channel-wise attention module is also employed to reduce redundant features of the feature pyramid network and provide refined results.

Introducing depth information in a suboptimal fusion strategy may have a negative influence on the performance of SOD. We discuss the advantages of the so-called progressive multi-scale fusion method and propose a mask-guided feature aggregation module (MGFA). We also introduce a mask-guided refinement module (MGRM) to complement the high-level semantic features and reduce the irrelevant features from multi-scale fusion, leading to an overall refinement of detection.

RGB-D methods sacrifice the model size to improve the detection accuracy, which may impede the practical application of SOD problems. To tackle this dilemma, we propose a dynamic distillation method along with simple noise elimination. To this end, the final model can significantly reduce the computational burden while maintaining the validity and mitigating the impact of distorted training data caused by low-quality depth maps.

Furthermore, in cosaliency detection, we propose a novel adaptive intra-group aggregation (AIGA) method to model the relationship between individual feature representation in a single image and group feature representation. This proposed AIGA can effectively improve the performance without increasing extra network parameters.

Copyright Declaration

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-Non Commercial 4.0 International Licence (CC BY-NC). Under this licence, you may copy and redistribute the material in any medium or format. You may also create and distribute modified versions of the work. This is on the condition that: you credit the author and do not use it, or any derivative works, for a commercial purpose. When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text. Where a work has been adapted, you should indicate that the work has been changed and describe those changes. Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

Acknowledgements

First and foremost, I would like to thank Dr.Tania Stathaki for her valuable suggestions and her guidance as well as her enthusiasm and continuous encouragement. Her skills and ability to explain complex concepts in an easy way helped me to develop an insight into analyzing, tackling and solving research problems. Tania is an extremely kind, caring and supportive advisor. She provides me with a free relaxed environment and makes me work without pressure. She always respects my decisions and supports my research. I would be grateful to her forever for providing me the opportunity that I can study as a Ph.D. student.

I am grateful to all the research colleagues, administrative and academic staff within the Communications and Signal Processing Group at the Department of Electrical and Electronic Engineering, Imperial College for providing an encouraging work environment.

I would like to thank my parents for their continual love and support. They always respect my decisions and fund my Ph.D. study. Without their support, I could hardly achieve what I have so far.

Dedication

I hereby declare that the content of this thesis is the product of my own research work. The main parts of this thesis have been published in the following list of my publications. I made the major contribution to design, implementation and experiments for these works under the guidance of Dr.Tania Stathaki and some others which have been acknowledged appropriately in text. The information, codes, and former ideas guided from other sources have been cited in a list of references is in the bibliography.

‘We cannot choose our external circumstances, but we can always choose how we respond to them.’

Epictetus

Contents

Abstract **3**

Copyright Declaration **5**

Acknowledgements **7**

1 Introduction **26**

1.1 Challenges of Visual Saliency Detection 27

1.2 Contributions 30

1.3 Thesis Outline 31

1.4 Publications 32

2 Foundations and Related Works **33**

2.1 Convolutional Neural Networks 33

2.1.1 Convolutional Layer 33

2.1.2 Pooling Layer 35

2.1.3 Activation Functions 36

2.2 Loss Functions 39

2.3 Optimization Algorithms 40

2.4 Weight Decay 44
2.5 Upsampling 45
2.6 Networks Using Blocks 46
2.7 Feature Pyramid Network 49
2.8 Multi-Scale Modules 50
2.9 Datasets and Evaluations 52
2.9.1 Datasets 52
2.9.2 Evaluations 57

3 Pyramid Self-Attention Module For Saliency Dectection 61

3.1 Introduction 61
3.2 Attention Mechanism 63
3.3 The Proposed Framework 64
3.3.1 Pyramid Self-Attention Module 64
3.3.2 Channel-Wise Attention 67
3.4 Evaluation 68
3.4.1 Impelmentation Details 69
3.4.2 Comparisons with State-of-the-arts 70
3.4.3 Ablation study 72
3.5 Discussion and Summary 73

4 Progressive Multi-scale Fusion Network for RGB-D SOD 75

4.1 Introduction 75
4.2 Related Work 78

4.2.1 RGB-D Salient Object Detection 78
4.2.2 Mask-guided Mechanisms 79
4.3 The Proposed Framework 79
4.3.1 The Overall Architecture 80
4.3.2 Mask-Guided Feature Aggregation 81
4.3.3 Mask-Guided Refinement Module 82
4.4 Experiments 84
4.4.1 Datasets 84
4.4.2 Evaluation Metrics 85
4.4.3 Implementation Details 85
4.4.4 Comparison with State-of-the-art Results 86
4.4.5 Ablation Studies 88
4.4.6 Discussion 93
4.5 Summary 93

5 Dynamic Knowledge Distillation for RGB-D SOD 95

5.1 Introduction 95
5.2 Knowledge Distillation 98
5.3 Methodologies 99
5.3.1 Overview 99
5.3.2 Dynamic Knowledge Distillation 100
5.3.3 Noises Elimination with the Dynamic Knowledge Distillation . 101
5.4 Experiments 104

5.4.1 Datasets and Evaluation Metrics . . . 104
5.4.2 Implementation Details . . . 104
5.4.3 Comparisons with the state of arts . . . 104
5.4.4 Ablation Studies . . . 107
5.5 Summary . . . 111

6 Adaptive Intra-Group Aggregation for Co-Saliency Detection 113

6.1 Introduction . . . 113
6.2 Related Works . . . 114
6.2.1 Co-Salient Object Detection . . . 114
6.2.2 Determinantal Point Processes . . . 114
6.3 Methodology . . . 115
6.3.1 Adaptive Intra-Group Aggregation . . . 115
6.3.2 Scale-aware Loss . . . 117
6.4 Experiments . . . 118
6.4.1 Datasets and Evaluation Metrics . . . 118
6.4.2 Implementation Details . . . 118
6.4.3 Performance Comparison . . . 119
6.5 Summary . . . 123

7 Conclusion 124

7.1 Summary of Thesis Achievements . . . 124
7.2 Potential Applications . . . 126
7.3 Future Work . . . 127

Bibliography **129**

List of Tables

1 Notations lookup table. 25

3.1 Quantitative results with F-score and MAE on five challenging datasets: DUTS-TE, ECSSD, HKU-IS, SOD and DUT-OMRON. Our method is compared with 11 competitive baseline methods. 70

3.2 Quantitative results with F-score and MAE on five challenging datasets: DUTS-TE, ECSSD, HKU-IS, SOD and DUT-OMRON for the different multi-scale modules. 72

3.3 Quantitative results with F-score and MAE on five challenging datasets: DUTS-TE, ECSSD, HKU-IS, SOD and DUT-OMRON for the different pyramid modules. 73

4.1 Quantitative comparison including the maximum of F_β, S_α, E_θ, and M, over five widely evaluated datasets. $\uparrow$ & $\downarrow$ represent higher and lower is better, respectively. The best two scores are highlighted in **red** and **blue**. 86

4.2 Ablation analyses on NLPR, LFSD and SIP. *Base* means an FPN network built on a VGG19 backbone. MGFA and MGRM are two proposed modules. ASPP is an Atrous Spatial Pyramid Pooling module. 91

4.3 Effectiveness of mask-guided mechanism on NLPR, LFSD. *Sum* and *Concat* indicate that we fuse RGB and depth features by employing summation and concatenation operation, respectively. 91

4.4 Quantitative comparison with different scales on NLPR, LFSD and SIP. 93

5.1 Quantitative comparisons through the maximum of F-score F_β, S-score S_α, E-score E_θ, and error-score M, over five widely evaluated datasets. DKD and $DKD*$ indicate simple VGG16-based and VGG19-based FPN respectively. *ite* refers to the training iterations. 106

5.2 The model size and inference speed of different methods. 106

5.3 Ablation analysis on 4 datasets. RGB and $RGBD$ indicate that the student network is trained without/with depth maps respectively by cross-entropy loss. s indicates the weight of knowledge distillation. Here we use the mean value of F-score F_m and wF_m to show the overall accuracy. 108

5.4 Quantitative comparisons of applying DKD on DANet. 110

5.5 Distillation on VGG16 with RGB maps. 111

6.1 Quantitative comparisons through the maximum of F-score F_β, S-score S_α, E-score E_θ, and error-score M, over three widely evaluated datasets. 119

6.2 Ablation studies on the proposed methods. 121

6.3 Comparison of GFLOPs and run-time (second) between our method and others. All models are evaluated on one Nvidia Titan XP, and the size of the input is 224×224×3. 121

6.4 Ablation studies on the scale-aware loss. 122

6.5 Comparisons of different diversity computation methods. 123

List of Figures

1.1 Relationships between different visual saliency detection models. . . . 27

1.2 Examples of saliency maps generated by simple FPN. False positive and negative issues appear in the predictions 28

1.3 Examples of RGB and depth maps 29

2.1 Two-dimensional cross-correlation operation. 34

2.2 Convolutional layer with multiple channels. The convolution process can be regarded as sliding the channel-wise filters on the input. Element-wise multiplication and addition are performed at the sliding locations to generate the outputs. Here the padding is 0 and the sliding stride is 2. 35

2.3 Maximum pooling with a pooling window shape of 2×2. 36

2.4 Classic neuron structure. 36

2.5 The Sigmoid activation function. 37

2.6 The Tanh activation function. 38

2.7 The Rectified Linear Unit (ReLU) activation function. 38

2.8 Bilinear interpolation operation. Four red dots are known points and the green dot is the point which needs to be interpolated. 45

2.9 A single VGG block. 46

2.10 A VGG network structure. 47

2.11 ResNet block with and without 1 × 1 convolution. 48

2.12 Dense connections in dense block. 48

2.13 Feature pyramid network. 49

2.14 Lateral connection operation in the top-down pathway. 50

2.15 Inception Module, naive version. 51

2.16 Pyramid Pooling Module 51

2.17 Atrous Spatial Pyramid Pooling. 52

2.18 Examples from five widely used testing datasets. 53

2.19 Examples of six widely used RGB-D datasets. 54

2.20 Examples from three widely used co-saliency datasets. 56

3.1 Qualitative visual results of ablation studies. (a) original images, (b) ground truth, (c) baseline, (d) baseline+self-attention (PSAM), (e) The proposed PSAMNet, including PSAM and channel-wise attention. 62

3.2 Overall pipeline of the proposed model: a pyramid self-attention module (PSAM) is built on the basic FPN structure and four channel-wise attention (CAM) modules are added in lateral connections. The numbers refer to the channel numbers of feature maps. 64

3.3 The structure of Pyramid Self-attention Module. The numbers indicate the channel numbers of feature maps. 65

3.4 stand-alone self-attention mechanism. 66

3.5 The structure of Channel-wise attention Module. 67

3.6 Results with PR curves on five benchmark datasets: DUTS-TE, ECSSD, HKU-IS, SOD and DUT-OMRON. The x-axis represents the recall rate and the y-axis represents the precision. 69

3.7 Overall comparison of qualitative visual results between our method and selected baseline methods. It shows that our method is capable to provide a more complete salient map and smooth boundaries. . . . 71

3.8 Heatmaps visualization of the baseline network without/with the PSAM. Four heatmaps on each row represent four different scales from deep to shallow layers. 73

4.1 Qualitative visual results of RGB and RGB-D based methods. (a) original image, (b) depth image, (c) the ground truth, (d) PoolNet [1],(d) PSAMNet [2] (Chapter 3), (f) CPFP [3], (g) our proposed method. . 75

4.2 Unstatisfactory predictions of existing methods. 77

4.3 The overall structure of our proposed network, where RGB and depth images are fed into two VGG19 backbones and aggregated progressively from deep to shallow layers. ASPP is the atrous spatial pyramid pooling module[4]. 80

4.4 The proposed mask-guided feature aggregation. $\hat{f}_r^i$ and f_d^i represent the i-th level of RGB and depth features, respectively. f_s^i is the output of previous MGFA $\hat{f}_s^{i+1}$. 81

4.5 The proposed mask-guided refinement module. 83

4.6 Predictions from four different levels of the proposed network. $Conv5-3$ represents the saliency map which is generated by f_r^5 and f_d^5. Saliency maps in the first row are generated solely by MGFA whereas the bottom row maps are produced by MGFA and MGRM. 84

4.7 Results with PR curves on five benchmark datasets: DUTS-TE, ECSSD, HKU-IS, SOD and DUT-OMRON. x-axis represents the recall rate and y-axis represents the precision. 87

4.8 Visual comparisons with state-of-the-arts in different scenarios, including complicated background, low-quality depth image, low contrast between foreground and background in RGB and depth and misleading objects. 88

4.9 Effectiveness of MGFA. *Base* represents the FPN baseline, *Add* means combining two-modal features by summation in the same two-stream framework. 89

4.10 Visual comparisons of the proposed modules. For each experiment we employ the corresponding modules on the framework. Especially, MGFA and MGRM are employed on five scales and the ASPP module is only applied at the end of the RGB stream. 90

4.11 Effectiveness of mask-guided mechanism in RGB-D. Three rows from top to bottom indicate different fusion methods, namely, summation, concatenation and mask-guided. Columns from left to right represent five stages from deep to shallow layers. 92

5.1 Comparison of three fusion strategies. 96

5.2 Weighted F-measure and model size on NLPR and SIP datasets. Our lightweight model with simple early fusion can achieve satisfactory detection results over different datasets. DKD and DKD* refer to the proposed dynamic knowledge distillation with VGG16 and VGG19 backbones respectively. 96

5.3 Our framework consists of two stages. We adopt a cumbersome network as the teacher model (top) and a feature pyramid network as the student model (bottom). 99

5.4 Low-quality depths in training. 102

5.5 Fm curves across four benchmarks on two different backbones. 105

5.6 Visual comparisons with existing methods 107

5.7 Teacher and student scatters with different distillation weights. We exhibit the last 10,000 training iterations and dynamic KD shows better overall performance. It is worth noting that using 0.3 suffers from the overfitting issue (Table 5.3). 109

5.8 Heatmap visual comparisons of different distillation weights: heatmaps from left to right indicate weights 0.3, 0.5, 0.7 and dynamic respectively. 110

5.9 Ablation studies of the proposed methods. *Baseline* represents that the student network is only trained on cross-entropy loss. DKD represents the proposed dynamic knowledge distillation and NE means noise elimination. 111

6.1 Overall structure of the proposed method. 116

6.2 Different sizes of objects from different categories. 117

6.3 F-measure curves on two public CoSOD datasets. 119

6.4 Qualitative comparisons with state-of-the-arts in different categories. 120

6.5 The left scatter shows the S values in the group-wise scale priors. The right shows the adaptive β_g values in the scale-aware loss. We analyze that the negligible difference in weights β_g leads to minor improvement. 122

7.1 Concealed objects and the corresponding ground truth 126

7.2 Polyp images and ground truth in medical segmentation. 127

7.3 Examples of an input image and the corresponding labels. 128

Table 1: Notations lookup table.

Notation	Description
α_t	The dynamic knowledge confidence of teacher model
α_d	Diversity between individual and group features
α_S	The Hyper-parameter in the evaluation S-measure
β_w	The Hyper-parameter in the weight decay
β_m	The Hyper-parameter in the Momentum optimization
β_F	The Hyper-parameter in the evaluation F-measure
β_s	The dynamic error rate of student model
B	MiniBatch
β_1, β_2	The hyper-parameters in Adam optimization
β_g	The parameter for group scales
S	The group scales
$\theta_{t,s}$	The dynamic weights for knowledge distillation
γ	The Hyper-parameter in the RMSProp
∂	The operation of partial derivatives
$\boldsymbol{g}$	Gradients
$\boldsymbol{w}$	Learnable weights
$\boldsymbol{v}$	Momentum
$\boldsymbol{s}$	The scale of learning rate in optimization
$\boldsymbol{Y}$	The subset of whole discrete set

Chapter 1

Introduction

Over the last decade, computer vision has achieved compelling progress due to the success of deep learning technology, which provides powerful feature representations compared to traditional methods. Salient Object Detection (SOD) is an important task in computer vision and has attracted more researchers to pay attention to it. In particular, this task simulates the human visual system to allocate more attention to attractive and interesting regions or objects for further processing. In other words, SOD aims at detecting or segmenting the most conspicuous objects in images.

The history of SOD is relatively short and traditional methods rely on hand-crafted low-level features to distinguish salient objects. The first deep-learning-based model can be traced to [5], which proposes a deep hierarchical structure based on a Visual Geometry Group (VGG) [6] and progressively refines the details of predictions by integrating the local context. However, only RGB images may provide insufficient information for complicated scenarios, such as different lighting conditions and low-contrast backgrounds. Therefore, depth information is introduced by [7] to provide complementary spatial information and boundary cues for RGB images. Fortunately, with the development of image devices, depth maps can be acquired easily because of the large availability of depth sensors.

As mentioned above, advanced image devices obtain image data conveniently, leading

to the explosive growth of data volume. This requires us to process multiple relevant images collaboratively. Under this circumstance, co-saliency detection stems from [8] and grows rapidly. As a novel branch of SOD, co-saliency detection refers to discovering the salient and common foreground from two or more relevant images. In general, two important properties should be satisfied by the co-saliency detection, i.e., (1) the object should be salient in each individual image, and (2) the object should be similar in appearance among multiple images. Compared to individual images, a group of images contains richer and more useful information such as intrinsic attributes of a specific category.

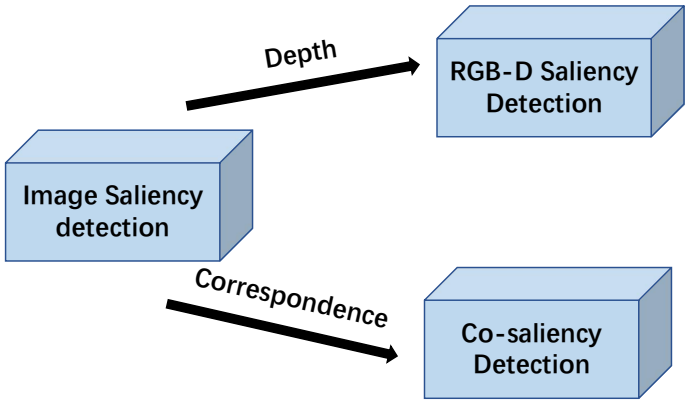


Figure 1.1: Relationships between different visual saliency detection models.

1.1 Challenges of Visual Saliency Detection

This saliency detection can be treated as a binary classification task on pixels. A robust saliency detector should be able to detect target objects with high confidence scores and clear contours in diverse scenarios. The challenges lie mainly in the following aspects.

RGB-based SOD.

SOD has achieved superior performance with the aid of deep learning networks [9]. However, as the networks become deeper, the deep-based methods are still restricted by some limitations. Generally, they utilize multiple convolution layers and pooling

layers to extract high-level semantic features. Multiple pooling operations can lead to information loss and degraded boundaries. In addition, when high-level features are upsampled to generate the score prediction for each pixel, the semantic features will be diluted while useless information will be introduced. As shown in Figure 1.2, the dilution of semantic features and the introduction of irrelevant multi-scale features cause false positives and negatives. In other words, besides the incomplete target objects, non-salient parts are also detected.

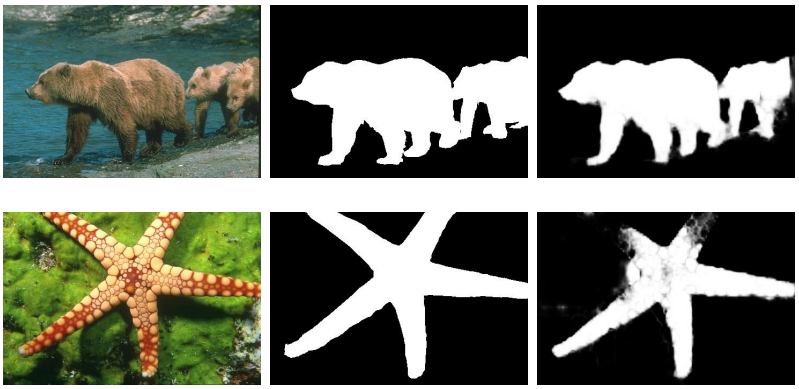


Figure 1.2: Examples of saliency maps generated by simple FPN. False positive and negative issues appear in the predictions

RGBD-based SOD.

RGB-D methods surpass the RGB methods in complicated scenes due to the utilization of depth maps, which provide spatial information and boundary cues from different domains as shown in the first row of Figure 1.3. However, using depth maps in an effective way is still a challenging issue. Concretely, there are three main problems in RGB-D methods. First, there are mainly three fusion strategies for the combination of RGB and depth maps, namely, early-stage fusion, multi-scale fusion and late-stage fusion. More specifically, early fusion combines RGB and depth maps in shallow layers and processes them together through the whole network structure. Multi-scale and late-stage fusion prefer to adopt two-stream structures, where one stream network is for extracting RGB features and another for depth features. Even though the effectiveness of these three fusion methods has been demonstrated, however, how to effectively exploit the depth information for feature

enhancement still needs to be investigated. Besides, the last two rows of Figure 1.3 exhibit the depth issue. Due to different depth sensors, the quality of the depth maps is unstable. Compared to the human depth map, the butterfly and flower are of low quality and unable to provide useful information. These low-quality depth maps may reduce the detection accuracy and how to avoid them in training also needs to be investigated.

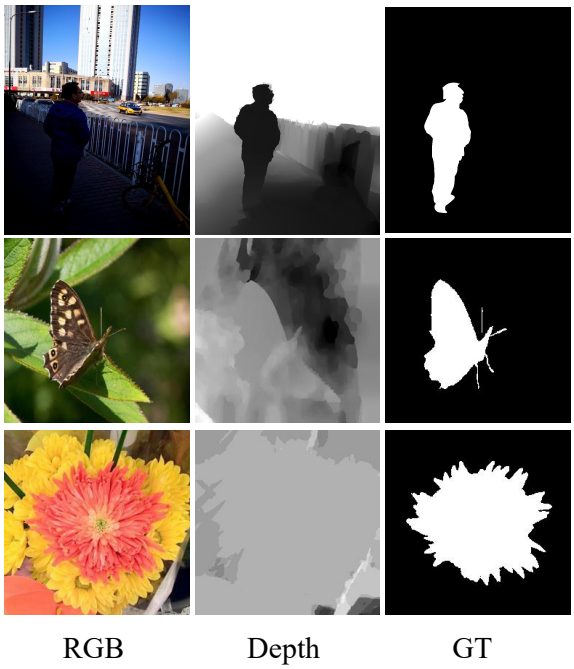


Figure 1.3: Examples of RGB and depth maps

Furthermore, the two-stream structures are more commonly exploited because they can achieve superior performance in most cases than the single-stream structure due to larger model sizes. Therefore, balancing detection accuracy and model size becomes an issue.

Co-saliency Detection.

Co-saliency detection introduces the inter-image correspondence as the common attribute constraint to discriminate the common objects from all the salient objects [8]. In order to explore the saliency representations from the perspectives of individual images and groups, intra- and inter-network structures are designed for processing different feature representations separately. However, how to effectively use the

individual and group features remains to be tackled.

1.2 Contributions

The primary contributions of the present work are the design and development of network modules and architectures. Besides, novel methods including training strategies and loss functions are also proposed. Notably, for fair comparisons, all training and testing datasets are the same as existing works and are available publicly. The contributions of this thesis can be summarised in the following three aspects:

- In order to remedy the false negative and positive issues in RGB SOD, we first propose a pyramid self-attention module (PSAM), which combines the attention mechanism with the multi-scale features and further enhances the high-level semantic features. Following the top-down pathway in the Feature Pyramid Network (FPN), enhanced features are provided to each scale to alleviate the dilution of semantic features. With the equipment of the proposed PSAM, the FPN is able to generate satisfied saliency maps. We further exploit channel-wise attention modules to filter the noise in multiple scales to refine the saliency maps.

- We propose to make full use of two-stream structures for RGB-D SOD. In order to alleviate the impact of low-quality depth maps, a mask-guided feature aggregation (MGFA) module is designed for effectively fusing cross-domain features and meanwhile filtering depth noise. Furthermore, due to the utilization of multi-scale fusion, irrelevant information can be introduced. Similar to the MGFA, we use the mask-guided mechanism to build a novel mask-guided refinement module (MGRM) to tackle this issue. Therefore, the final framework is able to progressively aggregate features from different domains and refine predictions from deep to shallow layers.

- To suppress the large model sizes and improve the inference speed in RGB-D SOD, we propose a novel dynamic knowledge distillation training strategy, which considers both the knowledge confidence from the teacher model and hard samples from the student model. Due to the utilization of a single stream structure and the proposed dynamic knowledge distillation, the final student model can achieve competitive performance in a smaller and faster network. We also eliminate the noise caused by distorted depth maps by taking full merit of the knowledge prior from the teacher model. Extensive experiments have been conducted to demonstrate the effectiveness of the proposed methods.

- Furthermore, inspired by the utilization of determinantal point processes in reinforcement learning, we propose an effective algorithm for co-saliency detection. This method can measure the discrepancy between individual and group features and aggregate these feature representations in an adaptive way. Considering the imbalance sizes between different categories, we also propose a scale-aware loss in the training phase. Extensive experiments have demonstrated that both methods can improve detection accuracy without increasing extra networks.

1.3 Thesis Outline

The rest of this thesis is structured as follows: Chapter 2 gives the fundamentals of deep learning and related networks in object detection and segmentation. Chapter 3 presents a novel pyramid self-attention module that introduces the attention mechanism to multi-scale feature maps for the enhancement of high-level features in SOD. Chapter 4 proposes a mask-guided feature aggregation and refinement module for cross-domain features fusion and saliency maps refinement. Chapter 5 considers the balance between detection accuracy and model size and proposes a novel dynamic knowledge distillation, which can effectively reduce the model size

and, meanwhile, achieve competitive performance. Chapter 6 takes advantage of the diversity computation of determinantal point processes to measure the discrepancy between the group and individual features and proposes an adaptive method to fuse the features. Chapter 7 draws conclusions for the thesis and provides potential research directions for future work.

1.4 Publications

The results of this research work have already been published or are currently under review:

- **Guangyu Ren**, Tianhong Dai, Panagiotis Barmpoutis, Tania Stathaki, "Salient Object Detection Combining a Self-Attention Module and a Feature Pyramid Network", Electronics 9 (10), 1702, 2020
- **Guangyu Ren**, Yanchu Xie, Tianhong Dai, Tania Stathaki, "Progressive Multi-scale Fusion Network for RGB-D Salient Object Detection", Computer Vision and Image Understanding, 103529, 2022
- **Guangyu Ren**, Yinxiao Yu, Hengyan Liu, Tania Stathaki, "Dynamic Knowledge Distillation With Noise Elimination for RGB-D Salient Object Detection", Sensors 22 (16), 6188, 2022
- **Guangyu Ren**, Tianhong Dai, Tania Stathaki, "Adaptive Intra-Group Aggregation for Co-Saliency Detection", ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).
- **Guangyu Ren**, Michalis Lazarou, Jing Yuan, Tania Stathaki, "Towards Automated Polyp Segmentation Using Weakly-and Semi-Supervised Learning and Deformable Transformers", is currently under review

Chapter 2

Foundations and Related Works

2.1 Convolutional Neural Networks

2.1.1 Convolutional Layer

Formally, in mathematics, the definition of convolution between two functions f and g is defined as:

$$(f * g)(x) = \int f(z)\, g(x - z)\, dz \tag{2.1}$$

Here, the function g is flipped and shifted by x. For discrete objects, the integral operation turns into a summation operation:

$$(f * g)(i) = \sum_{a} f(a)\, g(i - a) \tag{2.2}$$

In convolutional neural networks (CNNs), f refers to input and g refers to a convolution kernel, a filter, where the parameters are learned in the training phase. In the salient object detection task, the input is a $2D$ image with multiple channels.

Technically, this $3D$ input is also called a tensor in deep learning. It is worth noting that the actual convolution operation in deep learning is the cross-correlation operation, which is slightly different from the strict convolution. This is because since the kernels are learned from training data, the outputs of convolutional layers remain unaffected. Figure 2.1 exhibits the two-dimensional cross-correlation operation, the shaded output pixel is calculated by: $0 \times 0 + 1 \times 1 + 3 \times 2 + 4 \times 3 = 19$.

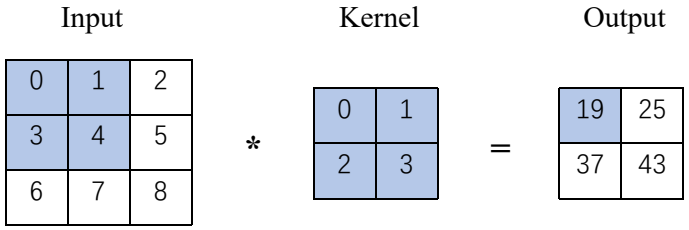


Figure 2.1: Two-dimensional cross-correlation operation.

Normally, the convolutional layer output in Figure 2.1 is called a feature map, as it can be regarded as the learned representations (features) in the spatial dimensions of the subsequent layer. In CNNs, the receptive field of each element refers to all elements which may affect its value.

In practice, the input is a $3D$ volume, namely, an RGB image with 3 channels. Figure 2.2 illustrates an example of multi-channel convolution. More specifically, blue volume refers to the input, and pink and green volumes refer to different filters (kernels) and their corresponding outputs, respectively. Typically, the shapes of the input and output volumes are $H_{in} \times W_{in} \times C_{in}$ and $H_{out} \times W_{out} \times C_{out}$. The kernel size is $H_k \times W_k \times C_k$. Here H, W and C represent the height, weight and the number of channels. The filters have the same number of channels as the input, namely, $C_k = C_{in}$ and the dimension of the output depends on the number of filters. Also, the amount of zero padding is P and the strides are S. The shapes between input and output have the following relationships:

$$H_{out} = (H_{in} - H_k + 2P)/S + 1 \tag{2.3}$$

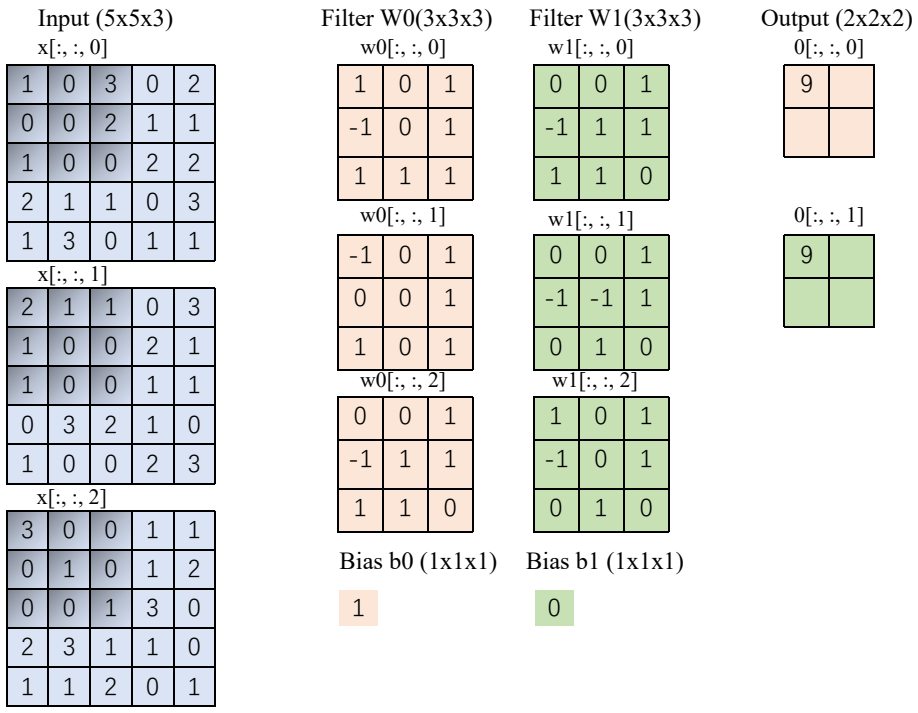


Figure 2.2: Convolutional layer with multiple channels. The convolution process can be regarded as sliding the channel-wise filters on the input. Element-wise multiplication and addition are performed at the sliding locations to generate the outputs. Here the padding is 0 and the sliding stride is 2.

$$W_{out} = (W_{in} - W_k + 2P)/S + 1 \tag{2.4}$$

In particular, convolutional neural networks commonly use kernels with odd height and width values, such as 1, 3, 5, or 7. Choosing odd kernel sizes has the benefit that we can preserve the spatial dimensionality while padding with the same number of rows on top and bottom and the same number of columns on left and right.

2.1.2 Pooling Layer

A pooling layer consists of a fixed-shape window that traverses all regions of the input with a stride. Typically, unlike the convolutional layer, the pooling layer contains no learnable parameters and calculates the maximum or average value in the pooling region. These pooling layers are called maximum pooling and average pooling respectively. Figure 2.3 shows the $2D$ maximum pooling operation and the

shaded element in the output is calculated by max (0,1,2,3).

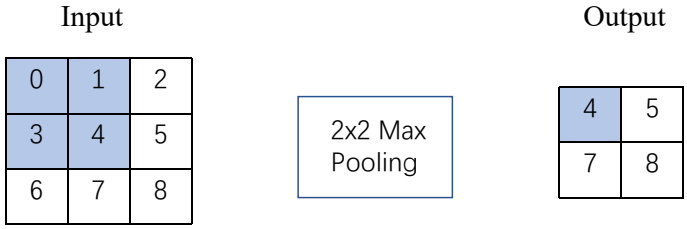


Figure 2.3: Maximum pooling with a pooling window shape of 2 × 2.

For multi-channel pooling layers, the number of output channels is the same as the input channels. In practice, the pooling layer is used to reduce the spatial dimensions and alleviate the excessive sensitivity of the convolutional layer to location.

2.1.3 Activation Functions

Nonlinear activation function is a key ingredient in deep neural networks. It is well-known that an affine function of an affine function is itself an affine function. In other words, even equipped with hidden layers, a combination of linear functions is still linear. Activation functions decide whether a neuron should be activated or not by calculating the weighted sum and further adding bias to it. Figure 2.4 illustrates that each convolutional layer is followed by a nonlinear activation function.

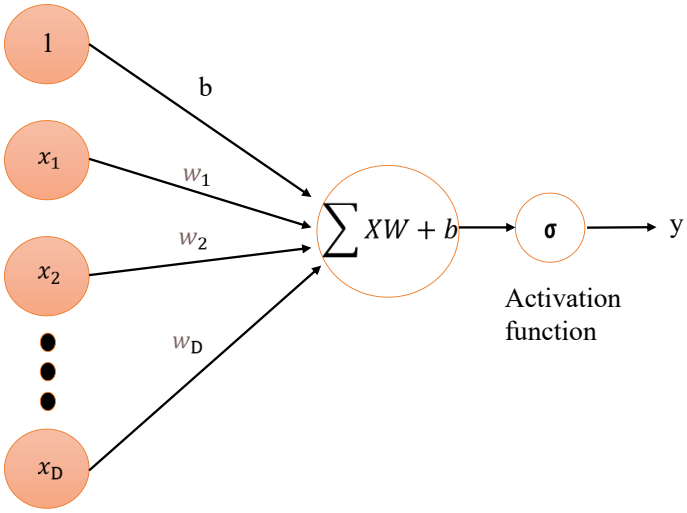


Figure 2.4: Classic neuron structure.

Sigmoid Function. The sigmoid function is a simple activation function that transforms the input value into the range $(0, 1)$. It can be expressed as:

$$\sigma(x) = \frac{1}{1 + exp(-x)} \tag{2.5}$$

The blue line and the red dashed line in Figure 2.5 exhibit the sigmoid function and its derivative, respectively. It can be observed that the derivative reaches a maximum of 0.25 at 0 point and approaches 0 from 0 in either direction. This property leads to the problem that the network updates parameters quite slowly when the gradients of hidden layers are small, leading to the so-called vanishing gradient.

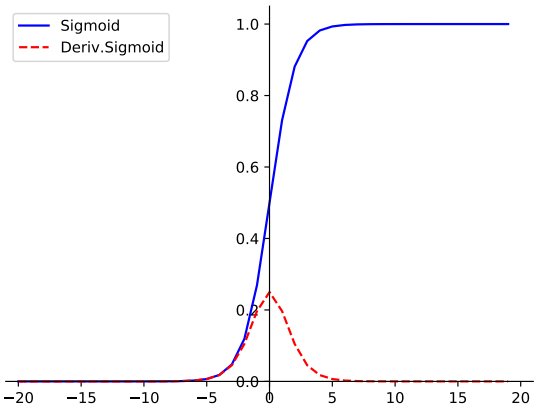


Figure 2.5: The Sigmoid activation function.

The sigmoid is still widely used as activation function to interpret the outputs as probabilities for binary classification problems.

Tanh Function. Tanh is a scaled version of the sigmoid whose mean value is 0. It squashes inputs to range $(-1, 1)$ as shown in Figure 2.6. It can be expressed as:

$$tanh(x) = \frac{exp(x) - exp(-x)}{exp(x) + exp(-x)} \tag{2.6}$$

Similarly to the sigmoid, the tanh also suffers from the vanishing gradient issue. Notably, this function is treated as a scale function in chapter 5.

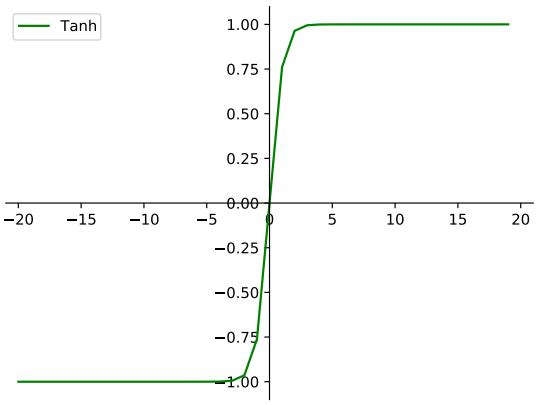


Figure 2.6: The Tanh activation function.

Rectified Linear Unit Function. Due to the simplicity of implementation and its good performance in a variety of predictive tasks, the Rectified Linear Unit (ReLU) becomes the most commonly used activation function in convolutional neural networks. ReLU provides a simple nonlinear transformation. Given an element x, the function is defined as the maximum of that element and 0:

$$\sigma(x) = max(0, x) \tag{2.7}$$

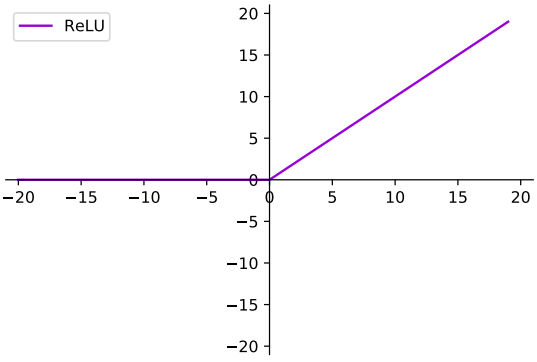


Figure 2.7: The Rectified Linear Unit (ReLU) activation function.

As illustrated in Figure 2.7, when the input is negative, the derivative of the ReLU function is 0, and when the input is positive, the derivative of the ReLU function is 1. Compared to the sigmoid function, the ReLU makes efficient computation in deep

neural networks and mitigates the problem of vanishing gradients.

2.2 Loss Functions

Kullback–Leibler Divergence.

Kullback–Leibler (KL) Divergence provides a way to measure if two distributions are close together or not. Formally, a random variable x following a probability distribution P with a probability density function (p.d.f.) $p(x)$ and an estimated probability distribution Q with $q(x)$. Then the KL divergence can be expressed as:

$$D_{KL}(P,Q) = \sum_{i=1}^{n} p(x_i) log \frac{p(x_i)}{q(x_i)} \tag{2.8}$$

In computer vision tasks, this KL divergence is commonly used for measuring the discrepancy between teacher and student models in knowledge distillation.

Cross Entropy.

The KL divergence can be transformed as:

$$D_{KL}(P,Q) = \sum_{i=1}^{n} p(x_i) log p(x_i) - \sum_{i=1}^{n} p(x_i) log q(x_i) \tag{2.9}$$

Here $-H(P) = \sum_{i=1}^{n} p(x_i) log p(x_i)$ is the information entropy, which is a constant. Therefore, the second term in equation 2.9 indicates the equivalent relationship:

$$H(P,Q) = -\sum_{i=1}^{n} p(x_i) log(q(x_i)) \tag{2.10}$$

This is the cross entropy loss, which is widely used in computer vision. Here, minimizing the cross entropy is equivalent to maximizing the log-likelihood function.

2.3 Optimization Algorithms

The objective function of the optimization algorithm is usually a loss function based on the training dataset, the goal of optimization is to reduce the training error. In other words, for a deep learning problem, the loss function evaluates the discrepancy between a prediction of a network and its corresponding ground truth from the training dataset. Optimization algorithms attempt to update model parameters and minimize the value of loss functions.

Gradient Descent.

First, iteratively computing the derivative (gradient) of the loss function L:

$$\nabla L = \frac{\partial L}{\partial \mathbf{w}} \tag{2.11}$$

Using the gradient $\frac{\partial L}{\partial \mathbf{w}}$ to update the initial weights $\mathbf{w}$ in the network:

$$\mathbf{w} \leftarrow \mathbf{w} - \eta \frac{\partial L}{\partial \mathbf{w}} \tag{2.12}$$

Here, the initial $\mathbf{w}$ and the constant learning rate $\eta > 0$ are chosen prior to training. Notably, the learning rate η plays a significant role in the deep learning model. If it is too small, the model hardly makes progress. If it is too big, the model may diverge. Therefore, in order to find a suitable learning rate, multiple experiments need to be carried out in practice.

In deep learning, the gradient is the average of all the examples in the training dataset. Given a training dataset of n examples, the gradient of loss function L is expressed as:

$$\nabla L = \frac{1}{n} \sum_{i=1}^{n} \frac{\partial L_i}{\partial \mathbf{w}} \tag{2.13}$$

Computer vision tasks such as image classification, segmentation and object detection require at least thousands of images with high resolutions for training. Therefore, the cost of gradient descent becomes higher with increasing training images.

Stochastic Gradient Descent.

In order to reduce the computational cost at each iteration, instead of computing on all examples, stochastic gradient descent (SGD) uniformly samples an index $i \in (1, ..., n)$ and computes the gradient to update parameters:

$$\mathbf{w} \leftarrow \mathbf{w} - \eta \frac{\partial L_i}{\partial \mathbf{w}} \tag{2.14}$$

Minibatch Stochastic Gradient Descent.

Gradient descent is inefficient for large-scale training data. Stochastic gradient descent is computationally inefficient since it cannot exploit the full power of computational resources. Under these circumstances, minibatch stochastic gradient descent is an elegant way to balance computational efficiency.

$$\mathbf{g} = \frac{1}{B} \sum_{i=1}^{B} \frac{\partial L_i}{\partial \mathbf{w}} \tag{2.15}$$

Where g is the gradient and B refers to a minibatch that is drawn uniformly at random from the training set. The batch size of B is a hyper-parameter in deep learning. More specifically, when the batch size equals the number of whole training data n, the minibatch stochastic gradient descent is gradient descent. When the batch size equals 1, the minibatch stochastic gradient descent is stochastic gradient descent. In practice, we pick a minibatch that is large enough to offer good computational efficiency while still fitting into the memory of a GPU. For instance, as suggested by [10, 11], we normally use batch sizes 4, 8 and 16 in salient object detection tasks on a single GTX TITAN X GPU.

Momentum.

Stochastic gradient descent suffers from noisy gradient due to instantaneous gradient. The momentum method is able to tackle this gradient descent problem by averaging gradients over the past. The gradient at time $t-1$ can be expressed as:

$$\mathbf{g}_{t,t-1} = \frac{1}{B}\sum_{i=1}^{B}\frac{\partial L_i}{\partial \mathbf{w}_{t-1}} = \frac{1}{B}\sum_{i=1}^{B}\mathbf{h}_{i,t-1} \tag{2.16}$$

Here using $\mathbf{h}_{i,t-1}$ represents the stochastic gradient descent for a sample i at time $t-1$. The momentum $\mathbf{v}_t$ is calculated by the leaky average over multiple past gradients:

$$\mathbf{v}_t = \beta_m \mathbf{v}_{t-1} + \mathbf{g}_{t,t-1} \tag{2.17}$$

$$\mathbf{w}_t \leftarrow \mathbf{w}_{t-1} - \eta \mathbf{v}_t \tag{2.18}$$

Where $\beta_m \in (0,1)$. This momentum method allows us to obtain more stable directions of descent. Especially, when $\beta_m = 0$ this method returns to the regular gradient descent.

Adaptive Gradient.

Typically, the learning rate decreases during the training phase. Adaptive gradient (Adagrad) provides a way for SGD to adjust the learning rate.

$$\mathbf{s}_t = \mathbf{s}_{t-1} + \mathbf{g}_t^2 \tag{2.19}$$

$$\mathbf{w}_t \leftarrow \mathbf{w}_{t-1} - \frac{\eta}{\sqrt{\mathbf{s}_t + \epsilon}}\mathbf{g}_t \tag{2.20}$$

Here using a variable $\mathbf{s}_t$ to accumulate past gradient variance. ϵ is an additive constant. Therefore, this method scales automatically with the magnitude of the gradients.

RMSProp.

On deep learning problems, Adagrad can sometimes be too aggressive in reducing learning rates, leading to the model converging slowly. To alleviate this issue, RMSProp is proposed [12] to use the leaky average on variable $\mathbf{s}_t$:

$$\mathbf{s}_t = \gamma \mathbf{s}_{t-1} + (1-\gamma)\mathbf{g}_t^2 \tag{2.21}$$

$$\mathbf{w}_t \leftarrow \mathbf{w}_{t-1} - \frac{\eta}{\sqrt{\mathbf{s}_t + \epsilon}}\mathbf{g}_t \tag{2.22}$$

Adam.

Adam incorporates advantages from RMSProp and momentum to adjust learning rates and gradients simultaneously. Both $\mathbf{v}_t$ and $\mathbf{s}_t$ is calculated:

$$\mathbf{v}_t = \beta_1 \mathbf{v}_{t-1} + (1-\beta_1)\mathbf{g}_t \tag{2.23}$$

$$\mathbf{s}_t = \beta_2 \mathbf{s}_{t-1} + (1-\beta_2)\mathbf{g}_t^2 \tag{2.24}$$

Here β_1 and β_2 are hyper-parameters and commonly set to $\beta_1 = 0.9$ and $\beta_2 = 0.999$ respectively [13]. Initializing $\mathbf{v}_0$ and $\mathbf{s}_0$ to 0. Note that normalization needs to be implemented on $\mathbf{v}_t$ and $\mathbf{s}_t$ due to a significant amount of bias initially:

$$\hat{\mathbf{v}}_t = \frac{\mathbf{v}_t}{1-\beta_1^t} \tag{2.25}$$

$$\hat{\mathbf{s}_t} = \frac{\mathbf{s}_t}{1 - \beta_2^t} \tag{2.26}$$

Therefore, the gradient is rescaled by $\hat{\mathbf{v}}_t$ and $\hat{\mathbf{s}}_t$:

$$\hat{\mathbf{g}_t} = \frac{\eta \hat{\mathbf{v}}_t}{\sqrt{\hat{\mathbf{s}_t}} + \epsilon} \tag{2.27}$$

$$\mathbf{w}_t \leftarrow \mathbf{w}_{t-1} - \hat{\mathbf{g}_t} \tag{2.28}$$

Adam uses bias correction to avoid slow startup and combines features of other optimization algorithms. Both Adam and SGD are commonly used in salient object detection.

2.4 Weight Decay

Deep neural networks are prone to raise the issue of overfitting, where the generalization gap between training and testing errors is large. In other words, the training error is significantly lower than the testing error. Thus, regularization techniques are commonly used in deep learning to combat the overfitting problem. In practice, the most simple and effective regularization way is weight decay. Here introducing a coefficient to reduce the weights when updating parameters in 2.28:

$$\mathbf{w}_t \leftarrow (1 - \beta_w)\mathbf{w}_{t-1} - \hat{\mathbf{g}_t} \tag{2.29}$$

Where the decay parameter β_w is set to 0.0005 [14]. Intuitively, the weight decay makes the deep models relatively degenerate to remedy the overfitting.

2.5 Upsampling

Deep CNNs gradually extract features and reduce the spatial dimensions to finally obtain feature maps with rich semantic information but low resolutions. In computer vision tasks, the feature maps generated by deep layers need to be upsampled and recovered to the same size as the original input images. Bilinear interpolation is usually applied to functions sampled on a 2D rectilinear grid. It is performed using linear interpolation first in one direction, and then again in the other direction as shown in Figure 2.8. Specifically, first we do the interpolation in the x direction:

$$f(x, y_1) = \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21}) \tag{2.30}$$

$$f(x, y_2) = \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22}) \tag{2.31}$$

Then applying interpolation in the y direction:

$$f(x, y) = \frac{y_2 - y}{y_2 - y_1} f(x, y_1) + \frac{y - y_1}{y_2 - y_1} f(x, y_2) \tag{2.32}$$

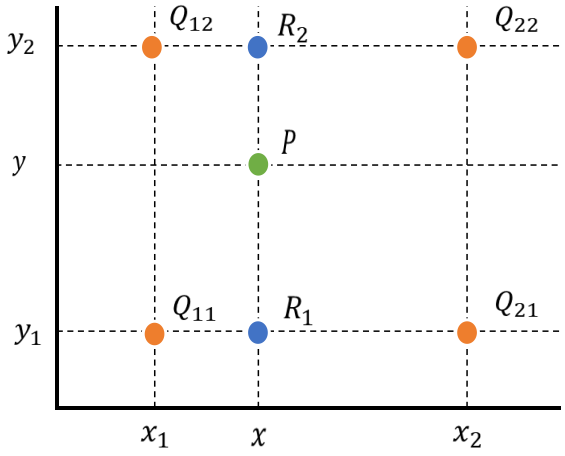


Figure 2.8: Bilinear interpolation operation. Four red dots are known points and the green dot is the point which needs to be interpolated.

2.6 Networks Using Blocks

Visual Geometry Group.

The idea of using blocks to build network structures arises from Visual Geometry Group (VGG) [6], which provides a general template to guide in designing novel networks.

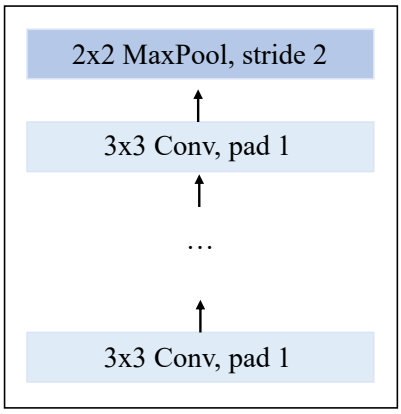


Figure 2.9: A single VGG block.

The VGG block as shown in Figure 2.9 employs convolutions with 3×3 kernels with padding of 1 and 2×2 max pooling with a stride of 2, which aims to half the resolution after each block. The VGG networks incorporate these repeated blocks to build backbones for extracting features in computer vision tasks. The VGG Network can be partitioned into two parts: the first consisting mostly of convolutional and pooling layers and the second consisting of fully-connected layers (in Figure 2.10).

The original VGG network has 5 convolutional blocks, among which the first two have one convolutional layer each and the latter three contain two convolutional layers each. It is called VGG-11 due to a total of 8 convolutional layers and 3 fully-connected layers. Note that VGG-16 and VGG-19 are commonly used in salient object detection tasks [9].

Residual Networks.

The residual connection mitigates the degeneration issue in deep neural networks. The residual block [15] follows the VGG's design and has two 3 × 3 convolutional

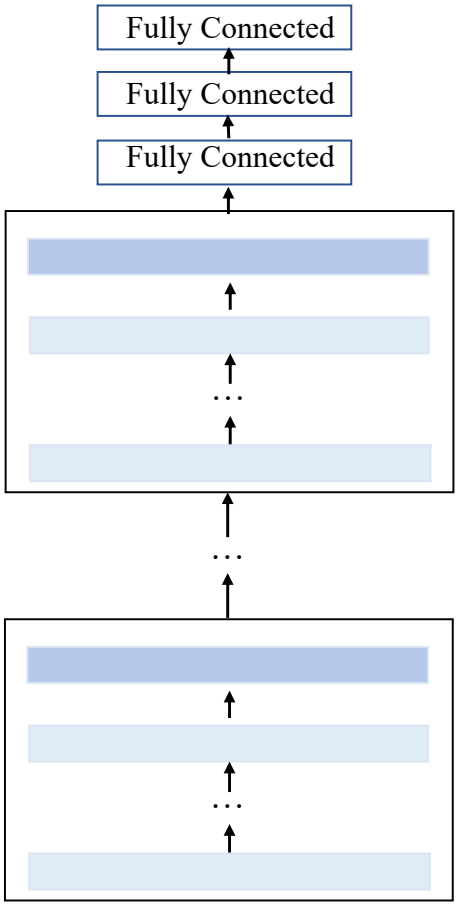


Figure 2.10: A VGG network structure.

layers with the same number of output channels. Each convolutional layer is followed by a batch normalization layer and a ReLU activation function.

Figure 2.11 exhibits two residual blocks. The 1×1 convolution is used for adjusting channel numbers. The residual network is built by stacking these blocks together with the first 7×7 convolutional layer and the final fully-connected layer. Note that ResNet-50 is commonly used as a backbone for feature extraction in salient object detection tasks [9].

Densely Connected Networks.

The main difference between residual block and densely connected block [16] is cross-layer connection, namely, addition and concatenation. Here we use $f_i()$ to represent the nonlinear transformation of $i-th$ layer and x_i to represent the output. The residual connection can be expressed as:

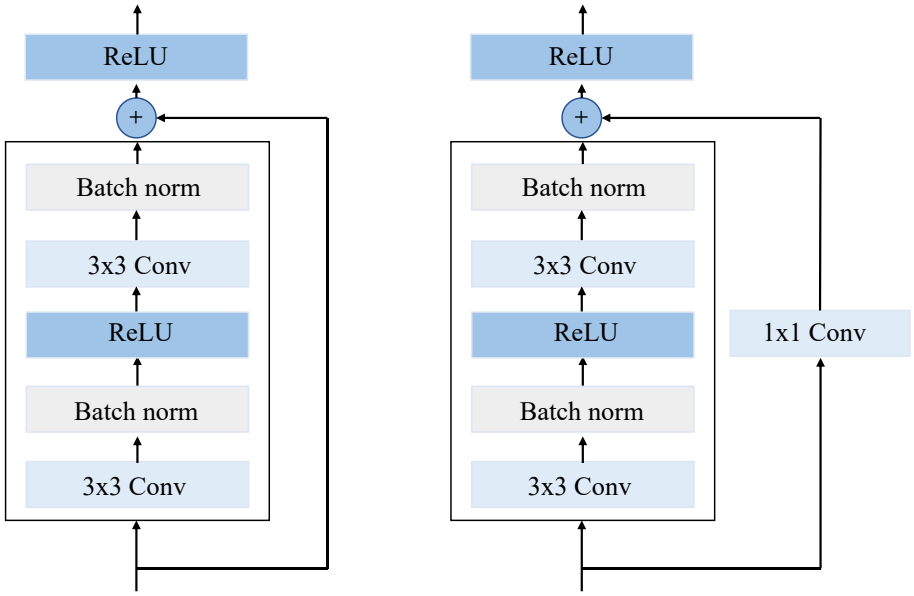


Figure 2.11: ResNet block with and without 1×1 convolution.

$$x_i = f_i(x_{i-1}) + x_{i-1} \tag{2.33}$$

The dense connection can be expressed as:

$$x_i = [f_i(x_{i-1}), x_{i-1}] \tag{2.34}$$

Where the $[,]$ indicates the concatenation operation.

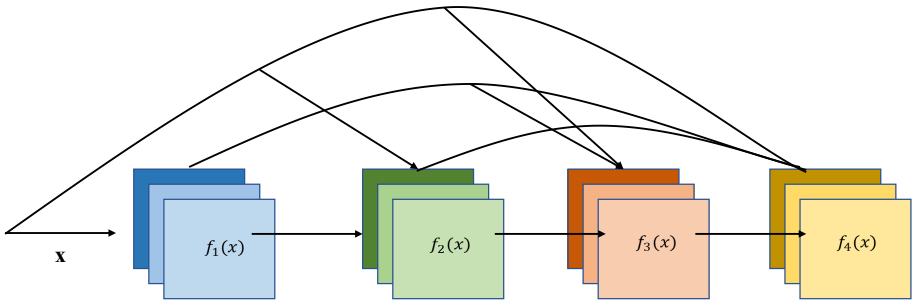


Figure 2.12: Dense connections in dense block.

This dense connection encourages feature reuse and makes full use of features from previous layers, leading to strengthened feature propagation. Normally, four dense layers are commonly used in practice as shown in Figure 2.12.

2.7 Feature Pyramid Network

Object detection and segmentation could benefit from multi-scale training and testing. Concretely, lower layers in CNNs capture rich spatial information while upper layers encode semantic knowledge. The main weakness of feature maps generated by shallow layers is short of rich semantic information compared to feature maps generated by deeper layers. This is because deep CNNs encode input images into feature maps through a hierarchical process, naturally causing the basic semantics to appear in the shallow layer feature maps, while the more complex semantics appear in the feature maps of deeper layers. SharpMask [17] proposes a top-down architecture to efficiently fuse spatial information from the lower layers and semantic information from the upper layers for object segmentation. This top-down architecture successively integrates the information from previous layers to refine the predictions.

Analogously, Feature Pyramid Network (FPN) [18] exploits the inherent multi-scale and pyramidal hierarchy of DNNs to construct feature pyramids for object detection. Both networks have a similar top-down structure with lateral connections as shown in Figure 2.13.

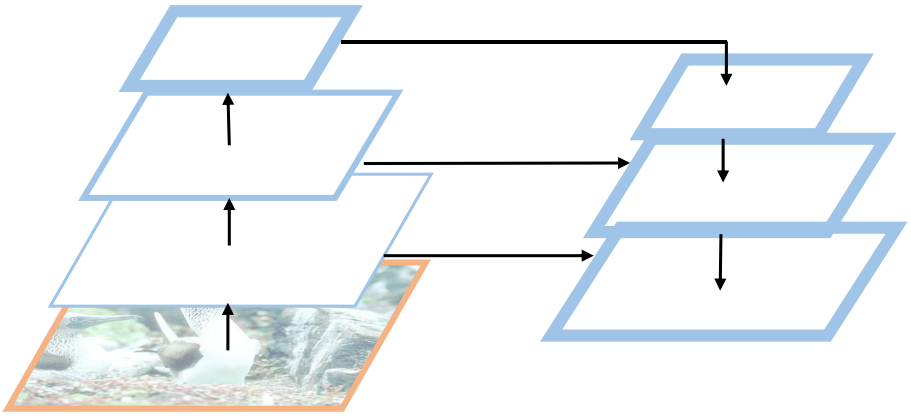


Figure 2.13: Feature pyramid network.

More specifically, the bottom-up pathway indicates the backbone network which extracts features at different scales from shallow to deep layers. The top-down structure transfers semantic information encoded by deeper layers to shallower layers through an upsampling operation to recover the spatial dimension. The upsampled

feature maps from deeper layers are merged with feature maps which are transformed under 1×1 lateral convolutional layers from shallower layers by element-wise addition (see Figure 2.14).

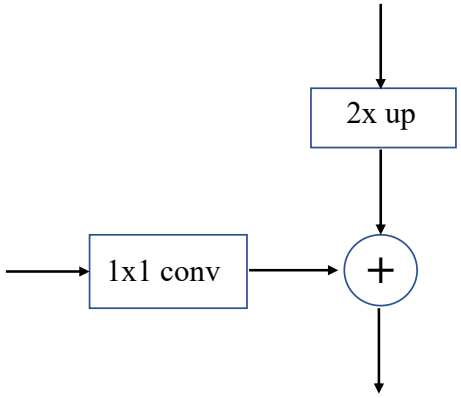


Figure 2.14: Lateral connection operation in the top-down pathway.

Thus, FPN is able to enhance the semantic representation capability of shallower layer feature maps by merging the semantic information encoded in deeper layer feature maps. This classic structure is widely used in object detection and segmentation tasks and is treated as the baseline framework in this thesis.

2.8 Multi-Scale Modules

In order to improve the performance of deep neural networks, the most straightforward way is to increase their sizes, including both depth and width. However, a large model size introduces a high computational cost and makes the network prone to overfitting. Instead of naively stacking convolutional layers, [19] follows the intuition that visual information should be processed at various scales and then aggregated so that the next stage can abstract features from different scales simultaneously. Unlike FPN, which makes full use of multi-scale features in a hierarchical way, the proposed Inception module in [19] is an important milestone in the development of CNNs.

Figure 2.15 exhibits the initial version of the inception module. It performs convolution on input, with 3 different sizes of filters, namely 1×1, 3×3 and 5×5.

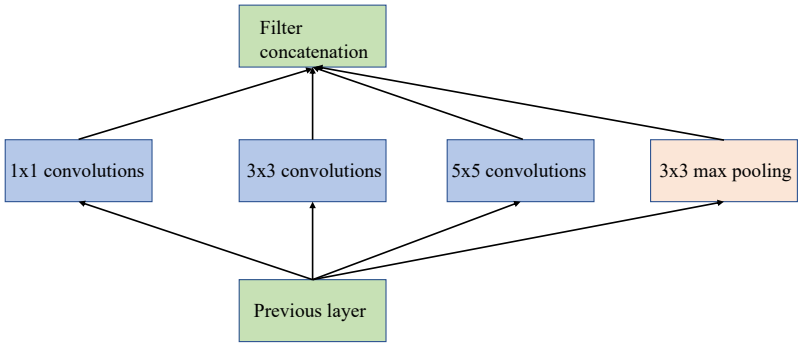


Figure 2.15: Inception Module, naive version.

Additionally, max pooling is also performed. The outputs are concatenated and sent to the next inception module. In order to limit the computational cost, 1×1 convolutional layers are added on each scale to reduce channel numbers.

Pyramid Pooling Module (PPM) [20] develops the multi-scale structure for scene parsing by fusing global context information and sub-region context for distinguishing among various categories.

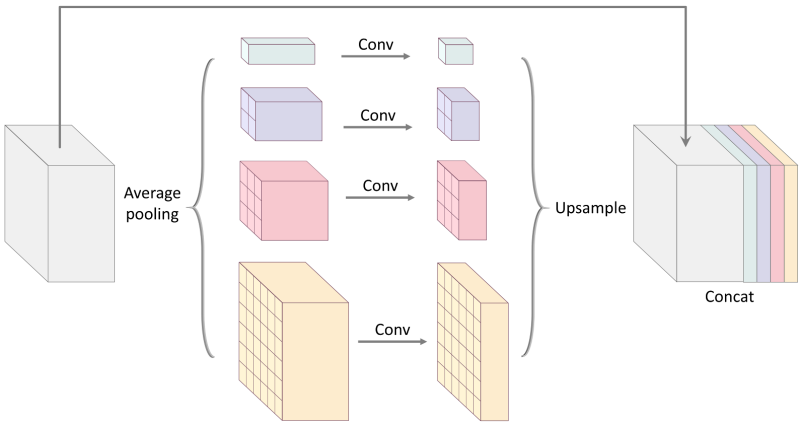


Figure 2.16: Pyramid Pooling Module

The original pyramid pooling module fuses features under four different pyramid scales. The coarsest level uses global pooling to generate a single bin output. The following pyramid level separates the feature map into different sub-regions and forms pooled representation for different locations. The output of different levels in the pyramid pooling module contains the feature map with varied sizes.

Atrous Spatial Pyramid Pooling (ASPP) [21] takes full merit of atrous convolution filters with a range of different dilation rates in parallel.

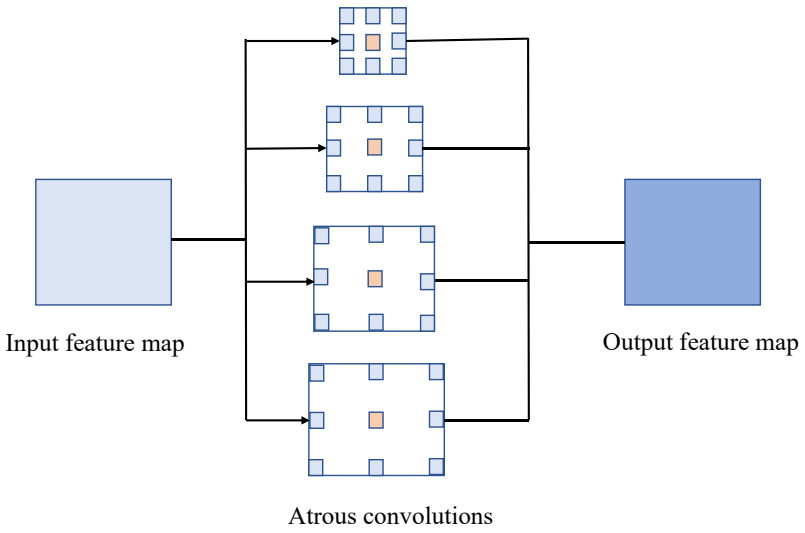


Figure 2.17: Atrous Spatial Pyramid Pooling.

Figure 2.17 illustrates that the initial ASPP obtains multi-scale feature representations by exploiting 3×3 atrous convolution with different dilated rates. Recently, a large number of analogous multi-scale modules have been proposed for feature enhancement in object detection and segmentation.

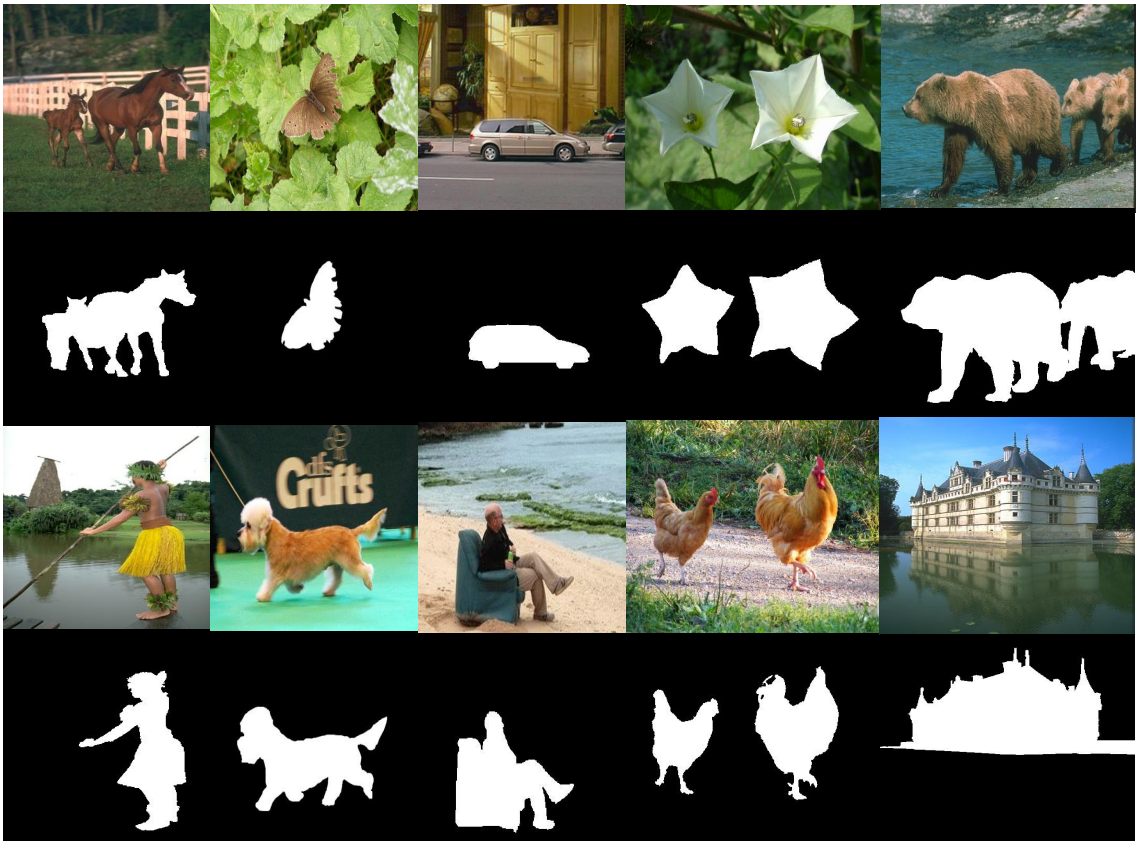
2.9 Datasets and Evaluations

2.9.1 Datasets

RGB Data

For RGB, five datasets consisting of a variety of objects are used for saliency detection, namely, ECSSD [22], DUT-OMRON [23], DUT-TE [24], HKU-IS [25] and SOD [26]. For the training of our model, we use the large-scale dataset DUTS [24], which contains 10,533 training images and 5,019 testing images (Figure 2.18). ECSSD is composed of 1,000 images with semantically meaningful but structurally complex natural contents. DUT-OMRON has 5,168 images of complex backgrounds and diverse content, with pixel-wise annotations. HKU-IS has 4,447 complex scenes that typically contain multiple disconnected objects with diverse spatial distributions and

similar fore-/background appearances. DUTS is a large-scale dataset, where the 10,553 training images are selected from the ImageNet [27] train/val set, and the 5,019 test images are from the ImageNet test set and SUN [28].



ECSSD DUT-TE DUT-OMRON HKU-IS SOD

Figure 2.18: Examples from five widely used testing datasets.

RGB-D Data

For RGB-D, we conduct our experiments on six publicly challenging RGB-D datasets: NJUD [29], NLPR [30], LFSD [31], DES [32], SIP [33] and DUT-RGBD [34] (Figure 2.19).

NLPR consists of 1,000 RGB images and their corresponding depth maps, which were obtained using a standard Microsoft Kinect. A series of outdoor and indoor locations are contained in this dataset, such as offices, supermarkets, campuses, and streets.

NJUD consists of 1,985 stereo image pairs, and these images were collected from

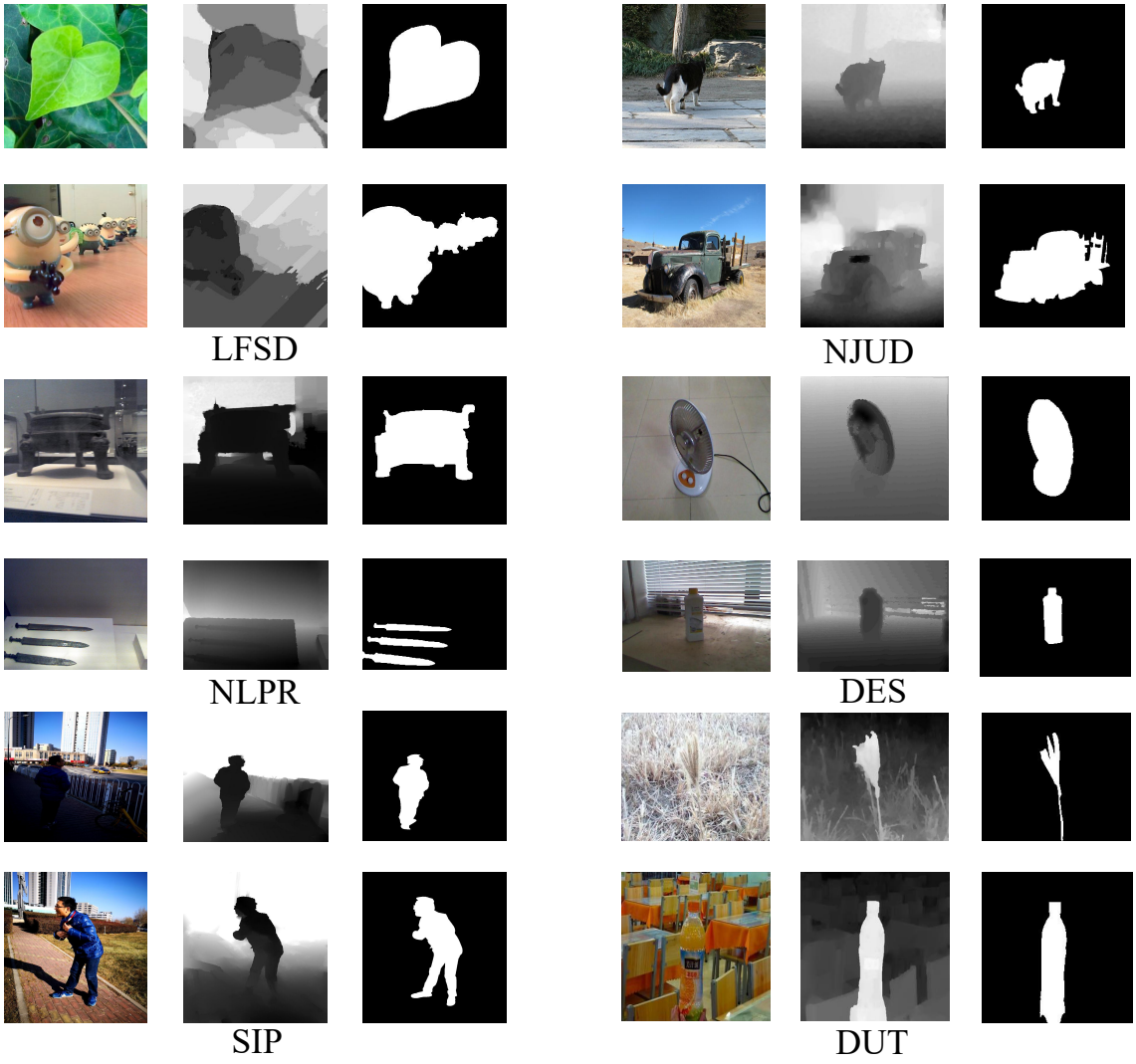


Figure 2.19: Examples of six widely used RGB-D datasets.

various sources, such as the Internet, 3D movies, and photographs that were taken with a Fuji W3 stereo camera.

LFSD includes 100 light fields collected using a Lytro light field camera, and 60 indoor and 40 outdoor scenes are contained respectively. It is worth noting that in order to label this dataset, three individuals were asked to manually segment salient regions, and only more than 90 percent of the overlapping areas in the three results are deemed final ground truth.

DES consists of 135 indoor RGB-D images with a resolution of 640 $\times$ 640, and these images were taken by Kinect. Similarly to LFSD, this dataset requires three individuals to label the salient object in each image, and use the overlapping area of

the three results as the final ground truth.

SIP is a large-scale testing dataset that consists of 929 annotated high-resolution images. A large number of images contain multiple people. In this dataset, depth maps were captured using a real smartphone (that is, Huawei Mate10). Besides, it is worth noting that this dataset covers diverse scenes and various challenging factors and is annotated with pixel-level ground truths.

DUT-RGBD consists of 800 indoor and 400 outdoor scenes with their corresponding depth images. This dataset includes several challenging factors, such as multiple or transparent objects, complex backgrounds, similar foregrounds and backgrounds, and low-intensity environments.

We adopt two different training datasets on two proposed RGB-D methods. Specifically, we use the same training dataset as [34] in chapter 4. This composited dataset contains 800 samples from DUT-RGBD [34], 1,485 samples from NJUD and 700 samples from NLPR. We adopt the same training strategy with [3] in chapter 5, where the training set contains 1,500 samples from NJUD and 700 samples from NLPR.

Co-saliency Data

For co-saliency detection, we adopt three current challenging datasets for evaluating the effectiveness: CoCA [11], CoSOD3k [35] and Cosal2015 [36] (Figure 2.20).

The Cosal2015 dataset is particularly established for the research in co-saliency detection. This dataset contains a total of 2,015 images, collected from the ILSVRC2014 detection benchmark [37] and the YouTube video set [38] and divided into 50 groups. The images in this dataset are highly cluttered. Moreover, in some cases, multiple salient objects are contained in every single image. These situations do not commonly appear in the image group, leading to a lot of ambiguity for the algorithms in determining which image regions are co-salient. Consequently, the Cosal2015 dataset appears to be much larger and more challenging than other existing datasets. In

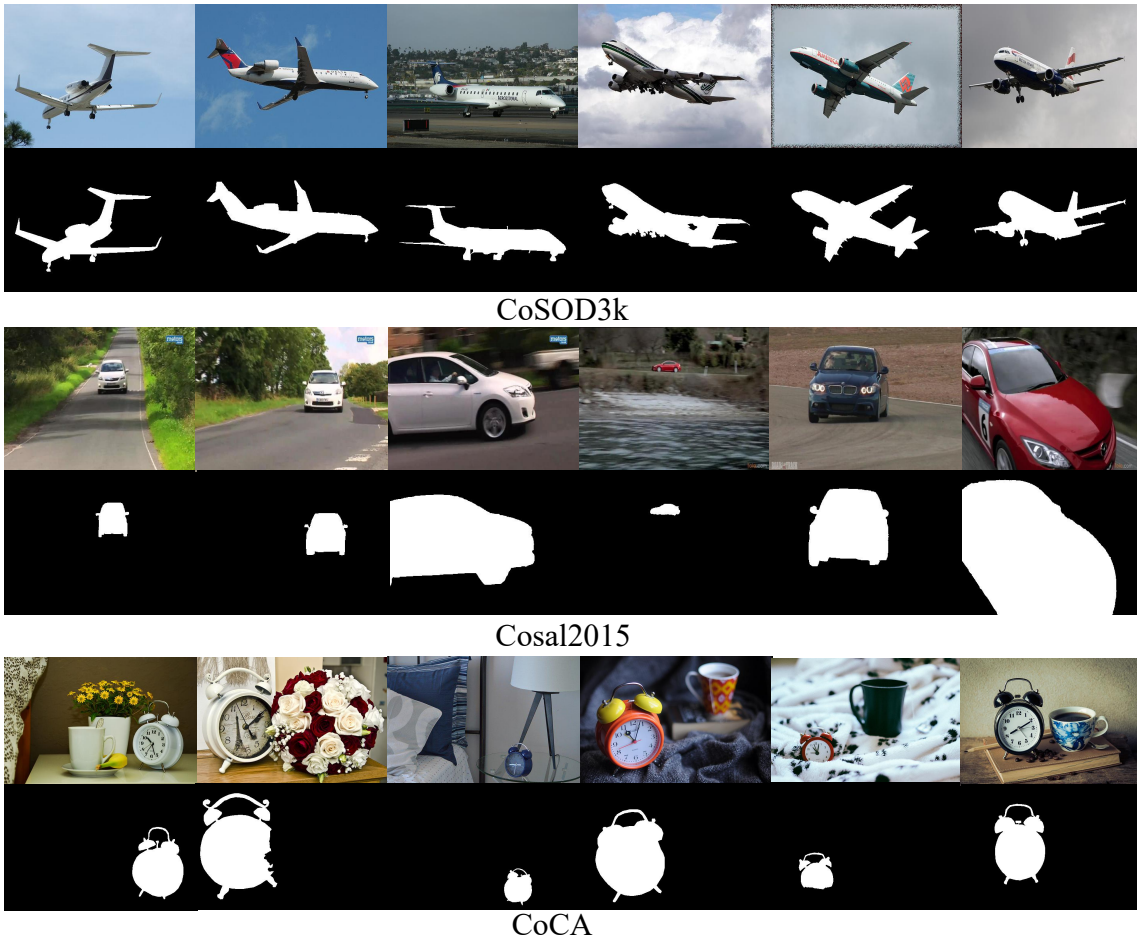


Figure 2.20: Examples from three widely used co-saliency datasets.

addition, it provides manually labeled pixel-level annotations by 20 subjects.

CoSOD3k is the largest Co-saliency detection dataset to date. More specifically, it divides 3,316 images into 160 groups and contains 13 super-classes, where each super-class is carefully selected in order to cover diverse scenes such as vehicle, food, tool, etc. In addition, each image is also equipped with hierarchical annotations to benefit diverse vision tasks. These annotations include category, bounding box, object, and instance.

CoCA dataset consists of 80 categories with 1,295 images. Diverse challenging scenarios are included in these images such as occlusion, clutter background, extraneous object interference, etc. The number of images in each category is different, varying from 8 to 40. This diversity is helpful in evaluating the ability of the model for different image set sizes. The number of co-salient instances in an image is also different.

Concretely, 336 images have more than two co-salient objects. The diversity of the number of instances can help to evaluate the robustness of the model to multi-object scenarios.

2.9.2 Evaluations

Here are several popular metrics for the evaluation of SOD, i.e., precision-recall (PR), F-measure [39], mean absolute error (MAE) [40], structural measure (S-measure) [41], and enhanced-alignment measure (E-measure) [42].

PR is calculated based on the binarized salient object mask and ground truth:

$$\text{precision} = \frac{TP}{TP + FP} \tag{2.35}$$

$$\text{recall} = \frac{TP}{TP + FN} \tag{2.36}$$

Where TP, TN, FP, and FN denote true-positive, true-negative, false-positive, and false-negative, respectively. The prediction is binarized by different thresholds from 0 to 255. Each threshold produces a pair of precision/recall values to form a PR curve for describing model performance.

F-measure indicates the overall performance by considering both precision and recall in a comprehensive way:

$$F_\beta = \frac{(1 + \beta_F^2) \cdot \text{precision} \cdot \text{recall}}{\beta_F^2 \cdot \text{precision} + \text{recall}} \tag{2.37}$$

Where β_F^2 is set to 0.3 as default. This generates a set of F-measure values for which previous methods report the maximal or average F_β. A general version of F_β is defined as:

$$F_{\beta}^{w} = \frac{(1+\beta_F^2) \cdot \text{precision}^w \cdot \text{recall}^w}{\beta_F^2 \cdot \text{precision}^w + \text{recall}^w} \tag{2.38}$$

Where w in F_{β}^{w} refers to different weights.

MAE measures the average pixel-wise absolute error between a predicted saliency map and its corresponding ground truth for all pixels, which can be defined by:

$$MAE = \frac{1}{W \times H} \sum_{x=1}^{W} \sum_{y=1}^{H} |P(x,y) - G(x,y)| \tag{2.39}$$

where W denotes the width and H denotes the height of the prediction, P denotes the prediction map which is the output of the model and G represents the ground truth map.

S-measure captures the importance of structural information in an image, S_α is used to assess the structural similarity between regional perception (S_r) and object perception (S_o). Thus, S_α can be defined by

$$S_\alpha = \alpha_S * S_r + (1 - \alpha_S) * S_o \tag{2.40}$$

Where α_S is empirically set to 0.5. Here S_r represents region similarity among k blocks by the structural similarity measure (SSIM):

$$S_r = \sum_{k=1}^{K} w_k * ssim(k) \tag{2.41}$$

Where w_k indicates different weights for each block. S_o represents foreground comparison O_{FG} and background comparison O_{BG}:

$$O_{FG} = \frac{2\tilde{x}_{FG}}{(\tilde{x}_{FG})^2 + 1 + 2\lambda * \sigma_{x_{FG}}} \tag{2.42}$$

$$O_{BG} = \frac{2\tilde{x}_{BG}}{(\tilde{x}_{BG})^2 + 1 + 2\lambda * \sigma_{x_{BG}}} \tag{2.43}$$

$$S_o = \mu * O_{FG} + (1 - \mu) * O_{BG} \tag{2.44}$$

Where $\tilde{x}_{\cdot}$ and $\sigma_{x_{\cdot}}$ represent the mean and standard deviation. μ equals the ratio of ground-truth area to the image area, namely $\mu = \frac{GT}{H \times W}$.

E-measure captures image-level statistics and their local pixel-matching information simultaneously. First, calculate the bias matrices for both the binary foreground map I_{FM} and the ground-truth map I_{GT}:

$$\varphi_{FM} = I_{FM} - \mu_{FM} \tag{2.45}$$

$$\varphi_{GT} = I_{GT} - \mu_{GT} \tag{2.46}$$

Where μ_{FM} and μ_{GT} indicate the global mean of the foreground and ground-truth maps. The alignment matrix ξ is defined as:

$$\xi_{FM} = \frac{2\varphi_{FM} * \varphi_{GT}}{\varphi_{FM} * \varphi_{FM} + \varphi_{GT} * \varphi_{GT}} \tag{2.47}$$

Finally, applying a mapping function on ξ_{FM} to build the enhancement alignment matrix θ_{FM} and E_θ:

$$\theta_{FM} = f(\xi_{FM}) \tag{2.48}$$

$$E_\theta = \frac{1}{W \times H} \sum_{x=1}^{W} \sum_{y=1}^{H} \theta_{FM}(x, y) \tag{2.49}$$

Where $f()$ is a quadratic function. Considering the popularity and characteristics of the existing metrics, F_β and MAE are more essential for evaluation, while other metrics are alternatively used in different works.

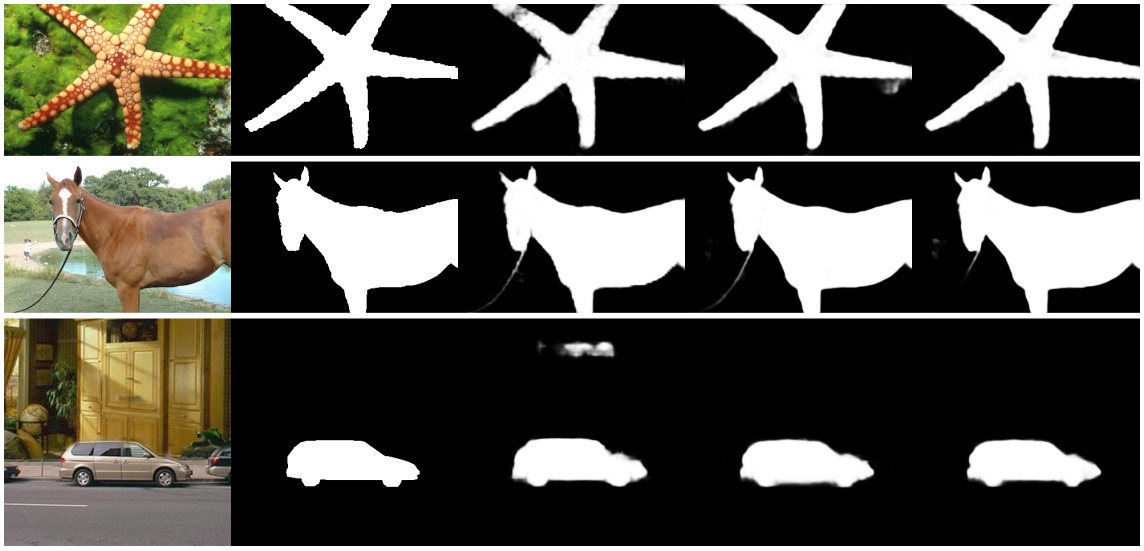
Chapter 3

Pyramid Self-Attention Module For Saliency Dectection

3.1 Introduction

Salient object detection or segmentation aims at identifying visually distinctive parts of a natural scene. With the capacity of providing high-level information, saliency detection problems are widely employed in various computer vision applications, such as object detection [43, 44, 45] and tracking [46], visual robotic manipulations [47, 48], image segmentation [49, 50] and video summarization [51, 52]. In early studies, salient object detection was formulated as a binary segmentation problem. However, the connections between salient object detection and other computer vision tasks were ambiguous. Nowadays, convolutional neural networks (CNNs) attract more attention in the research community. Compared with the classic hand-crafted feature descriptors [53, 54], CNNs have robust feature representation ability. Specifically, CNN kernels with small receptive fields can provide local information and the kernels with large receptive fields can provide global information. This characteristic enables CNN-based approaches to detect salient areas with refined boundaries [55]. Thus, CNN-based approaches have, in previous years, become the major research approach

for salient object detection problems.



(a) Image (b) ground truth (c) baseline (d) +PSAM (e) PSAMNet

Figure 3.1: Qualitative visual results of ablation studies. (a) original images, (b) ground truth, (c) baseline, (d) baseline+self-attention (PSAM), (e) The proposed PSAMNet, including PSAM and channel-wise attention.

Furthermore, the so-called Fully Convolutional Networks (FCNs) have recently become the fundamental framework for the above-mentioned problems [1, 56, 57], as FCNs can achieve richer spatial information compared to the fully connected layer for the same input size. However, although these works have achieved great improvement in performance, they are still restricted by some limitations. FCN-based approaches utilize multiple convolution layers and pooling layers to extract high-level semantic features. However, when the high-level features are upsampled to generate the score prediction for each pixel, the features will also be diluted which could decrease the ability of object localization.

In this thesis, we propose a novel pyramid self-attention module (PSAM) to overcome the limitation of feature dilution of the previous FCN-based approaches. Figure 3.1(c) shows the inherent problems of Feature Pyramid Networks (FPNs). By incorporating a self-attention module with multi-scale feature maps of FPNs, the model will focus on high-level features. This leads to the extraction of features with richer high-level semantic information and larger receptive fields. In addition, a channel-wise attention module is employed to eliminate the redundancy in the FPN which can refine the

final results. Experimental results show that the proposed PSAM can improve the performance of salient object detection and achieve superior state-of-the-art results in five challenging datasets. To summarise, the contributions of this work can be concluded as: 1) We propose a novel pyramid self-attention structure that can make the model focus more on the high-level features and reduce feature dilution in top-down pathways. 2) We adopt channel-wise attention to reduce the redundant information in the lateral connections of the FPN to refine the final results.

3.2 Attention Mechanism

Attention mechanism, an approach to solve the problem of loss of relevant information, is extensively used in the area of Natural Language Processing (NLP). [58] introduced a framework called transformer, a new encoder-decoder architecture that uses only the attention mechanism instead of recurrent layers to encode each position to capture global dependencies between input and output. This framework also allows parallel computing which leads to faster computational speed compared to recurrent networks. Except for the sequence models, this kind of attention mechanism is also needed in CNN models. Different from the attention mechanism in the sequence models, the self-attention mechanism pays more attention to a single data. [59] proposed Attention Augmented Convolutional Network which produces attentional feature maps via a self-attention module and concatenates these with CNN feature maps to capture spatial dependencies of the input and it achieves huge improvement in the tasks of object classification and detection. The stand-alone self-attention layer was introduced in the work of [60]. It can be used to set up a fully self-attentional model by replacing all spatial convolution layers with self-attention layers. The previous works prove that it can be used as an effective stand-alone layer that can replace the spatial convolution layer easily.

3.3 The Proposed Framework

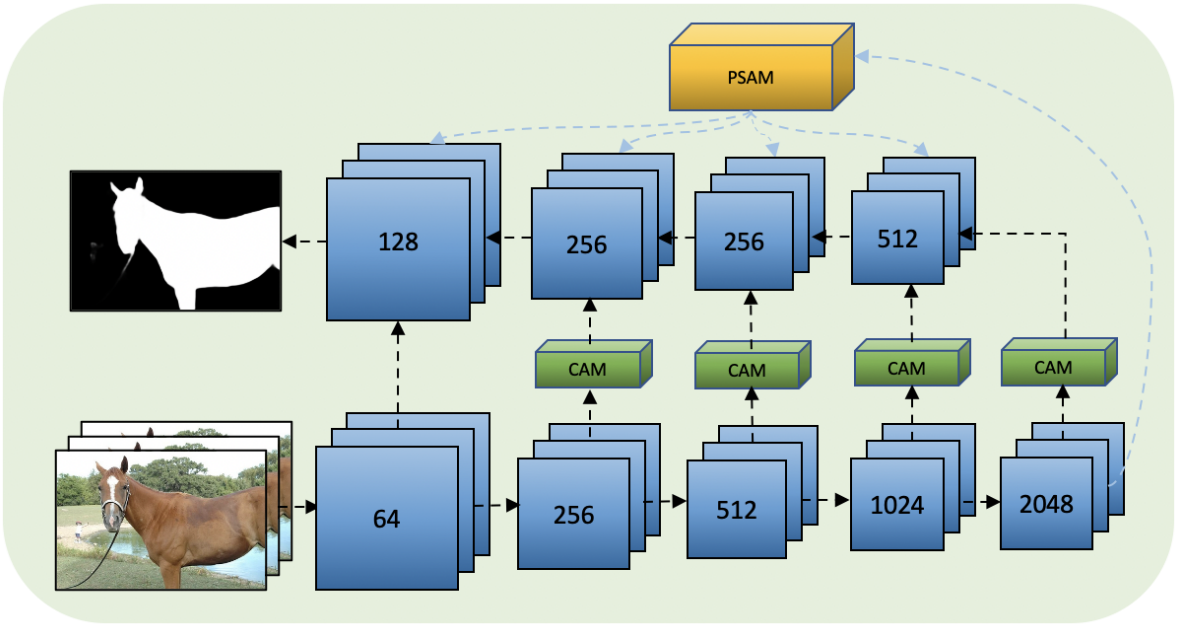


Figure 3.2: Overall pipeline of the proposed model: a pyramid self-attention module (PSAM) is built on the basic FPN structure and four channel-wise attention (CAM) modules are added in lateral connections. The numbers refer to the channel numbers of feature maps.

In this section, we describe the proposed novel architecture that integrates the two attention modules. More specifically, we use a pyramid self-attention module that aims to enhance the high-level semantic features and transmit the enhanced semantic information to different feature levels. In addition, when feature maps are merged in the top-down pathway, a simple channel-wise attention [61, 62, 63] module is added in each lateral connection to focus on the salient objects in feature maps.

3.3.1 Pyramid Self-Attention Module

In this subsection, we describe the proposed module framework in detail and demonstrate the differences from previous works. [64] has demonstrated that high-level semantic features are more representative and discriminative, leading to the position of salient objects being located more accurately. Figure 3.1(c) shows that without any additional attention modules, the FPN baseline can generate rough saliency

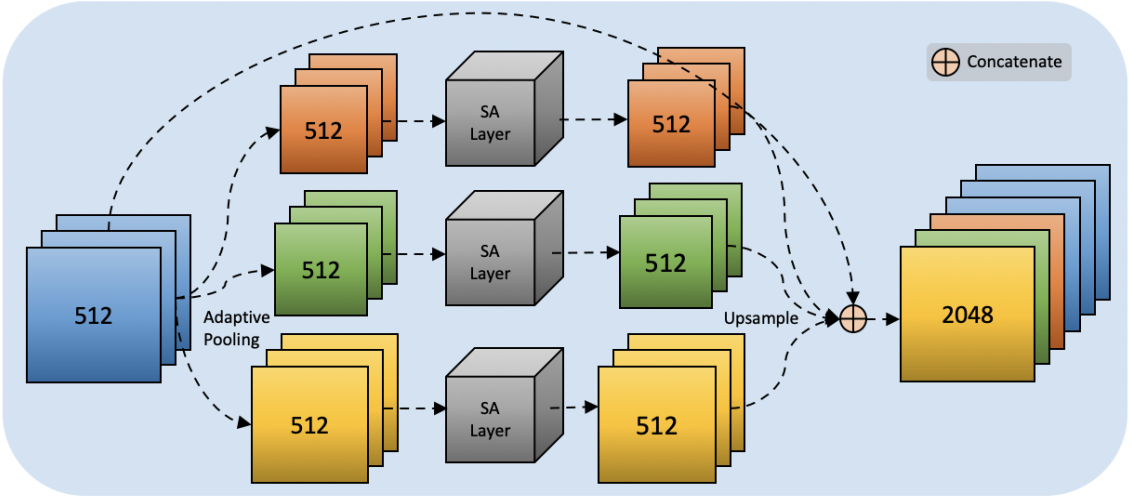


Figure 3.3: The structure of Pyramid Self-attention Module. The numbers indicate the channel numbers of feature maps.

maps which have incomplete salient objects. Meanwhile, there are also some non-salient objects present that should not be detected in the saliency maps. These error predictions are caused by two main challenges which cannot be avoided in the FPN architecture. The first problem is that the high-level information is diluted progressively when it is integrated with different feature levels in the top-down pathway. Another problem of the FPN baseline is that this architecture can be impacted by other non-essential information which may reduce the final performance of the model. In other words, the FPN architecture detects not only incomplete salient objects but also irrelevant objects. To overcome these two intrinsic problems of the baseline, we propose a novel pyramid self-attention module (PSAM) that contains stand-alone self-attention layers [60] in different scales, further focusing on important regions and enlarging the receptive field of the model. Specifically, as shown in Figure 3.3, PSAM firstly transforms the feature map which is produced by the bottom-up pathway into multi-scale feature regions and then each self-attention layer learns to pay more attention to important semantic information. After processing by self-attention layers, these multi-scale representations, which contain effective semantic information, are concatenated together to complement high-level semantic information in the top-down pathway. More technically, let X^{in} denote the feature map, which is produced by the topmost layer. We downsample the feature map

$X^{in} \in R^{H\times\ W\times C}$ into three different scales denoted as $\{X_1, X_2, X_3\}$, where H, W and C represent the height, width and channels of the feature map respectively. Given a pixel $x_{ij} \in \{X_1, X_2, X_3\}$ a corresponding local memory block r_k^{ij} is extracted from the same feature map $X_i \in \{X_1, X_2, X_3\}$. This r_k^{ij} is a $k \times k$ region which surrounds x_{ij}. There are three crucial learnable parameters in this self-attention algorithm, namely, queries, keys and values. We use W_Q, W_K and W_V to represent their learnable weights respectively. The final attention output pixel is computed as follows:

$$s_{ij} = \sum_{a,b\in N_k} \text{softmax}(q_{ij}^T k_{ab}) v_{ab}, \tag{3.1}$$

where $q_{ij} = W_Q x_{ij}$, $k_{ab} = W_K x_{ab}$, $v_{ab} = W_V x_{ab}$ denote the three crucial parameters, s_{ij} denotes the output pixel of a self-attention layer, $N_k = \{(a,b) : a = \{-\frac{k-1}{2}, ..., \frac{k-1}{2}\}, b = \{-\frac{k-1}{2}, ..., \frac{k-1}{2}\}\}$ defines the coordinates x_{ab} of r_k^{ij}. Figure 3.4

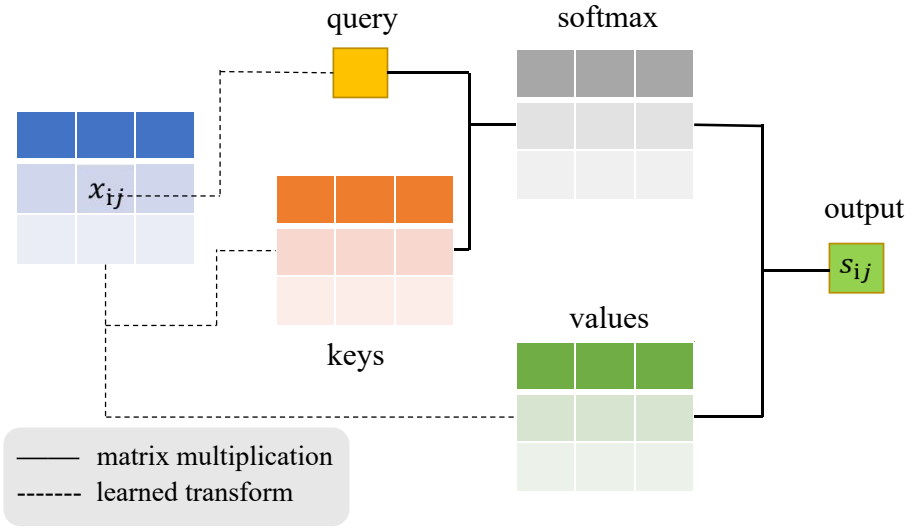


Figure 3.4: stand-alone self-attention mechanism.

exhibits the stand-alone self-attention mechanism, where the similarity between the query and x_{ab} is modulated by the content. Similarly to convolution operations, this local self-attention aggregates spatial information over neighborhoods. In addition, self-attention further mixes softmax weights, which are parametrized by content interactions. We use $\{Y_1, Y_1, Y_3\}$ to denote the final feature map which further represents an upsampling operation after each self-attention layer. Then we concatenate

them with the original X^{in} to generate the final output of the PSAM.

$$Y^{out} = \text{concat}\{Y_1, Y_1, Y_3, X^{in}\} \tag{3.2}$$

3.3.2 Channel-Wise Attention

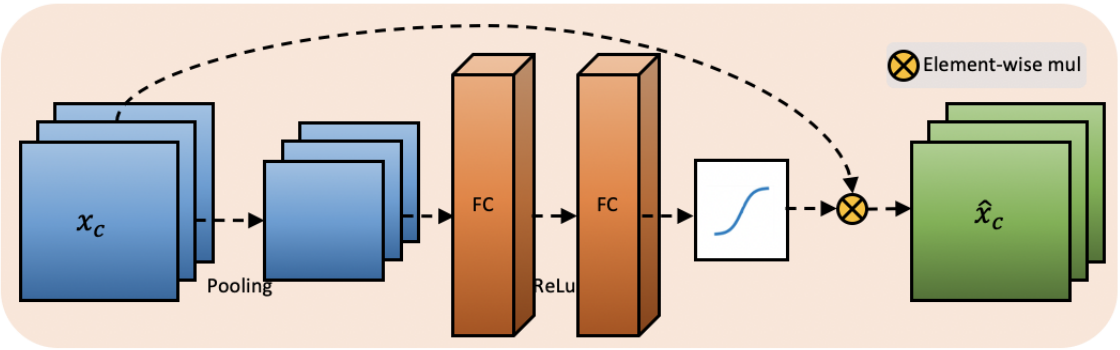


Figure 3.5: The structure of Channel-wise attention Module.

To enhance the contextual and structural information, the lateral connection has been used in the top-down pathway, leading to a state-of-the-art performance of detection tasks. However, this operation also introduces some unmeaningful information, which can reduce the performance and impact the final prediction. From Figure 3.1(c), the two problems caused by feature redundancy are obvious. The first problem is that there are extra regions that should not be detected in the saliency map. Another problem is that the edges of salient objects are ambiguous. Both problems indicate that further refinement should be applied. [61] has pointed out that different channels have different semantic features and channel-wise attention can capture channel-wise dependencies. In other words, channel-wise attention can emphasize the salient objects and alleviate the inaccuracy which is caused by redundant features in channels. Therefore, we add this simple channel-wise attention [61, 62, 63] to each lateral connection to achieve a refinement task. The structure of channel-wise attention is shown in Figure 3.5. It consists of one pooling layer and two fully-connected layers which are followed by a ReLU [65] and a sigmoid function respectively. First, an operation of squeezing global spatial information is applied to

each channel. This step can be easily implemented by an average pooling :

$$p_c = F_{pool}(X_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_c\,(i, j) \tag{3.3}$$

where c refers to the channel number and H x W refers to the spatial dimensions of the i-th element of X_c. After the pooling operation, the generated channel descriptor is fed into the fully-connected layers to fully capture channel-wise dependencies

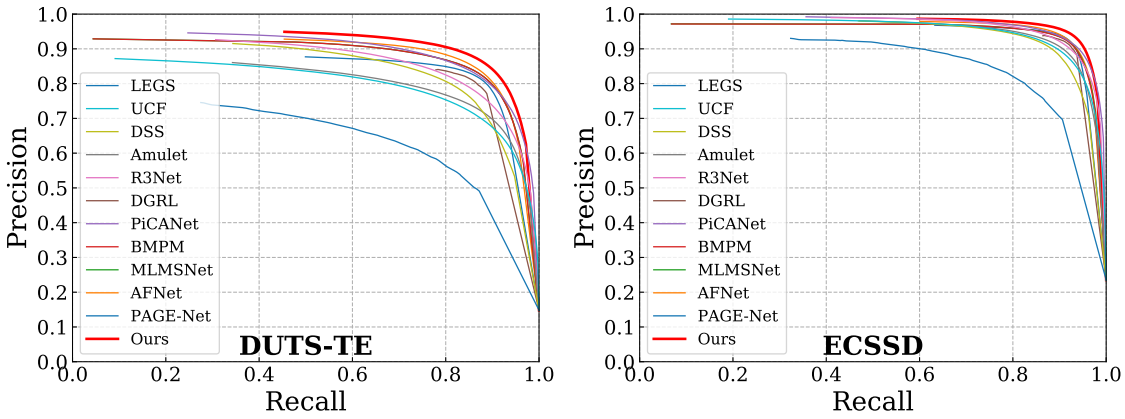
$$s_c = F_{fc}(p_c, W) = \sigma(fc_2(\delta(fc_1(p_c, W_1)), W_2)) \tag{3.4}$$

where σ refers to the sigmoid function, δ refers to the Rectified Linear Unit (ReLU) function and fc denotes the fully-connected layers. Finally, this generated scalar s_c multiplies the feature map X_c to generate a weighted feature map $\hat{X}_c$:

$$\hat{X}_c = s_c \cdot X_c \tag{3.5}$$

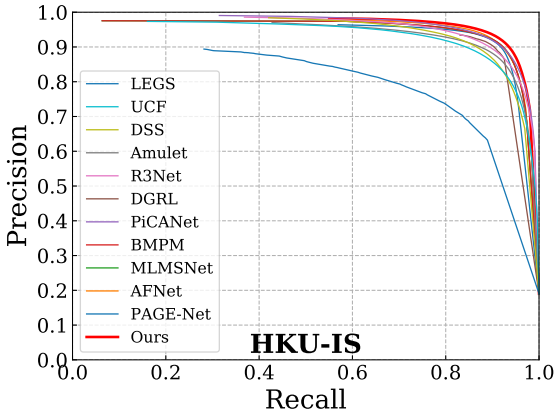
3.4 Evaluation

For the evaluation of the proposed methodology, we carry out a series of experiments using five popular saliency detection benchmarks, namely, ECSSD [22], DUT-OMRON [23], DUT-TE [24], HKU-IS [25] and SOD [26]. These five datasets consist of a variety of objects and structures which are challenging for salient object detection algorithms to locate and detect precisely. For the training of our model, we use the large-scale dataset DUTS [24], which contains 10,533 training images and 5,019 testing images. To evaluate the performance of the model, we estimate three representative evaluation metrics, namely, precision-recall (PR) curves, F-score (2.37) and mean absolute error (MAE) (2.39).

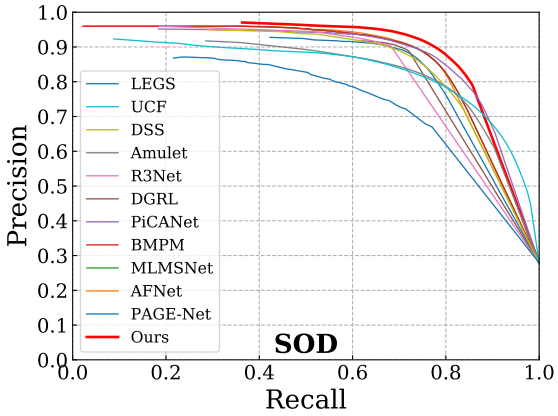


(a) DUTS-TE

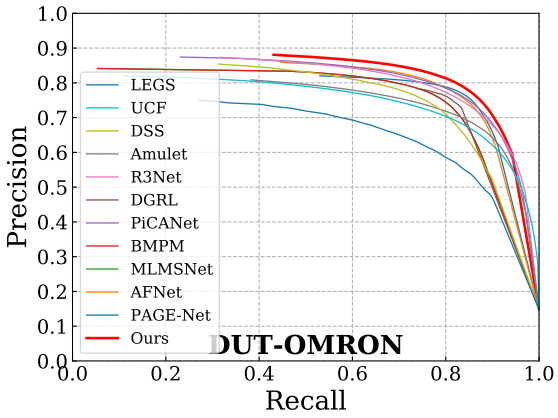
(b) ECSSD



(c) HKU-IS



(d) SOD



(e) DUT-OMRON

Figure 3.6: Results with PR curves on five benchmark datasets: DUTS-TE, ECSSD, HKU-IS, SOD and DUT-OMRON. The x-axis represents the recall rate and the y-axis represents the precision.

3.4.1 Impelmentation Details

Our model is implemented in Pytorch. We use ResNet-50 [15] as a backbone which has been pre-trained on ImageNet [66]. The proposed architecture is trained on

Method	DUTS-TE		ECSSD		HKU-IS		SOD		DUT-OMRON	
	F-score	MAE	F-score	MAE	F-score	MAE	F-score	MAE	F-score	MAE
LEGS [67]	0.654	0.138	0.827	0.118	0.770	0.118	0.733	0.196	0.669	0.133
UCF [68]	0.771	0.117	0.910	0.078	0.888	0.074	0.803	0.164	0.734	0.132
DSS [64]	0.813	0.064	0.907	0.062	0.900	0.050	0.837	0.126	0.760	0.074
Amulet [69]	0.778	0.085	0.914	0.059	0.897	0.051	0.806	0.141	0.743	0.098
R3Net [70]	0.824	0.066	0.924	0.056	0.910	0.047	0.840	0.136	0.788	0.071
PiCANet [71]	0.851	0.054	0.931	0.046	0.922	0.042	0.853	0.102	0.794	0.068
DGRL [72]	0.828	0.050	0.922	0.041	0.910	0.036	0.845	0.104	0.774	0.062
BMPM [73]	0.851	0.048	0.928	0.045	0.920	0.039	0.855	0.107	0.774	0.064
PAGE-Net [56]	0.838	0.051	0.931	0.042	0.920	0.036	0.841	0.111	0.791	0.062
MLMSNet [74]	0.852	0.048	0.928	0.045	0.920	0.039	0.855	0.107	0.774	0.064
AFNet [75]	0.863	0.045	0.935	0.042	0.925	0.036	0.856	0.109	0.797	0.057
PoolNet [1]	**0.880**	0.040	0.944	0.039	**0.933**	**0.032**	0.870	**0.101**	0.803	0.056
Ours	0.879	**0.040**	**0.944**	**0.038**	0.931	0.034	**0.874**	0.104	**0.813**	**0.056**

Table 3.1: Quantitative results with F-score and MAE on five challenging datasets: DUTS-TE, ECSSD, HKU-IS, SOD and DUT-OMRON. Our method is compared with 11 competitive baseline methods.

a GTX TITAN X GPU for 24 epochs. We follow the training settings and data agumentation in [1], the initial learning rate is set equal to 5e-5 for the first 15 epochs and then reduces to 5e-6 for the last 9 epochs. We adopt a 0.0005 weight decay for the Adam [13] optimizer and binary cross entropy loss function in the proposed framework. Finally, to increase the robustness of the model, we perform data augmentation through the application of random horizontal flipping.

3.4.2 Comparisons with State-of-the-arts

We perform our proposed method on five datasets to compare with 12 previous state-of-the-art methods, which include LEGS [67], UCF [68], DSS [64], Amulet [69], R3Net [70], DGRL [72], PiCANet [71], BMPM [73], MLMSNet [74], AFNet [75], PAGE-Net [56] and PoolNet [1]. For fair comparisons, we use the results which are generated by their original work with default parameters and released by the authors. Furthermore, all results are evaluated by the same evaluation method without any other processing tools. It is worth noting that this chapter was finished in 2020, therefore we also provide some current state-of-the-art methods, such as [76, 77, 78, 79].

Quantitative Comparisons

Figure 3.6 and Table 3.1 show the evaluation results of the proposed framework in comparison to eleven state-of-the-art methods on five challenging salient object datasets as already mentioned above. More specifically, in Figure 3.6, it is clearly demonstrated from the PR curve that the proposed methodology represented by the red line outperforms the state-of-the-art methods. This result means that our method has better robustness than other previous methods. Furthermore, the quantitative results are listed in Table 3.1. The proposed method achieves higher F-score and lower error scores than other methods, demonstrating that our novel model outperforms almost all previous state-of-the-art models on the different testing datasets. Although it is noted that PoolNet has similar state-of-the-art results, our model has 23% fewer parameters.

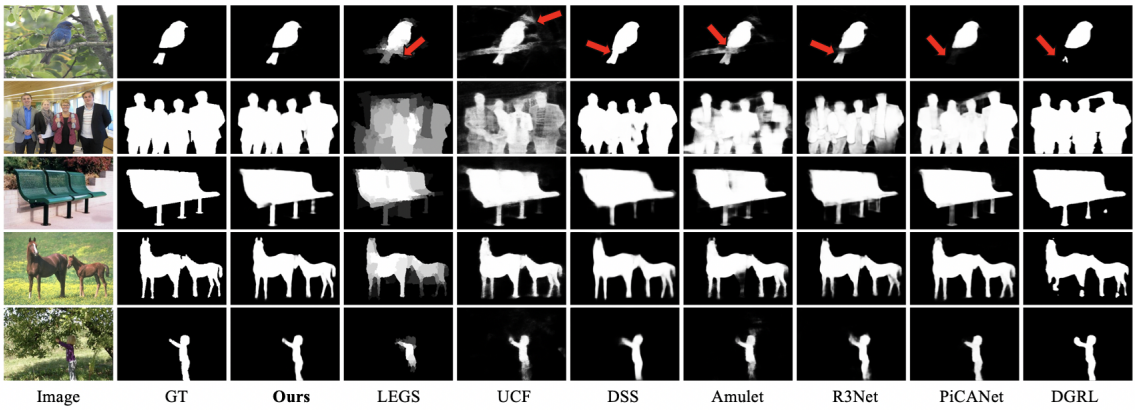


Figure 3.7: Overall comparison of qualitative visual results between our method and selected baseline methods. It shows that our method is capable to provide a more complete salient map and smooth boundaries.

Qualitative Comparisons

Figure 3.7 illustrates the visual comparisons to further show the advantages of the proposed method. More precisely, compared to other approaches, the proposed method demonstrates the best performance in different challenging scenarios, with the results being closer to the corresponding ground truths compared to previous methods.

3.4.3 Ablation study

In this subsection, we conduct a series of experiments on five different datasets to investigate the effectiveness of two modules. The ablation experiments are trained on DUTS [24] training dataset in the same environment. From Table 3.2, the model which contains PSAM and channel-wise attention module achieves the best performance, demonstrating that the proposed modules can effectively assist the baseline's salient object detection performance. More specifically, we initially conduct the baseline experiments on the FPN baseline with ResNet-50 as the backbone. This basic model can generate rough saliency maps for example the ones depicted in Figure 3.1(c). Furthermore, we add a pyramid self-attention module (PSAM) on the baseline and the F-score increases significantly on all benchmark datasets, an observation which is more prominent for DUTS-TE [24] and SOD [26]. On this basis, we add channel-wise attention to the model to compose the proposed framework. The final results show that the channel-wise attention modules can further increase performance and alleviate error predictions. To this end, Figure 3.1(d) and (e) demonstrate the effectiveness of the two modules respectively.

Method	DUTS-TE		ECSSD		HKU-IS		SOD		DUT-OMRON	
	F-score	MAE	F-score	MAE	F-score	MAE	F-score	MAE	F-score	MAE
Baseline	0.856	0.045	0.933	0.045	0.921	0.037	0.848	0.116	0.785	0.059
+PSAM	0.876	0.041	0.940	0.042	0.928	0.034	0.857	0.121	0.803	0.056
PSAM+CA	**0.879**	**0.040**	**0.944**	**0.038**	**0.931**	**0.034**	**0.874**	**0.104**	**0.813**	**0.056**

Table 3.2: Quantitative results with F-score and MAE on five challenging datasets: DUTS-TE, ECSSD, HKU-IS, SOD and DUT-OMRON for the different multi-scale modules.

We further visualize the feature maps of different scales from deep to shallow layers. The top row of Figure 3.8 illustrates that without PSAM, the baseline structure is unable to learn discriminative feature representations in deep layers and the overall feature maps have lower scores on each pixel compared to the bottom row. Obviously, with the aid of PSAM, discriminative features can be provided on each scale even the deepest feature map, which shows a rough object shape.

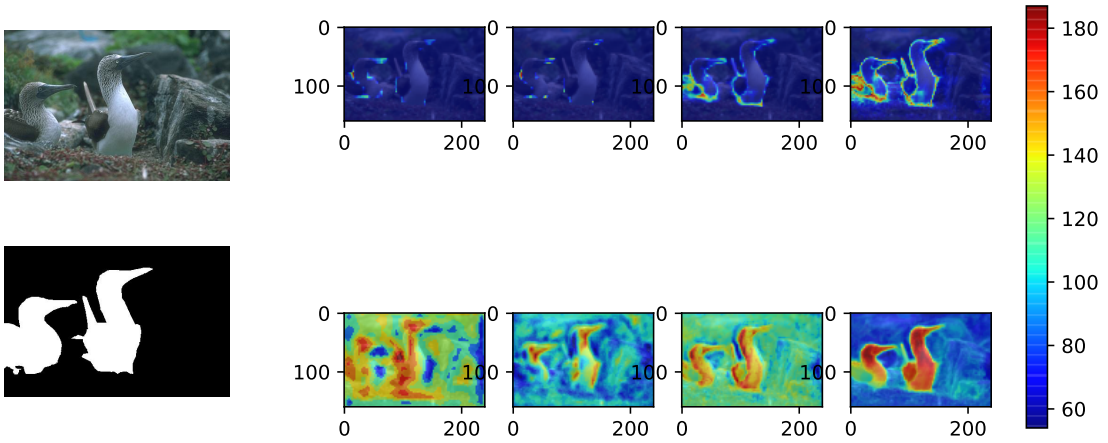


Figure 3.8: Heatmaps visualization of the baseline network without/with the PSAM. Four heatmaps on each row represent four different scales from deep to shallow layers.

3.5 Discussion and Summary

As shown in Figure 3.1(d), (e) and Table 3.2, it is observed that the proposed PSAM could enhance the semantic features and help the model focus on the useful objects. Table 3.3 further shows the superiority of the proposed PSAM compared to the existing pyramid pooling module (PPM) and atrous spatial pyramid pooling (ASPP) module. We believe that this attention module could provide further help to other computer vision tasks, such as object detection and segmentation, especially for semantic segmentation, which could facilitate multi-object saliency detection.

Method	DUTS-TE		ECSSD		HKU-IS		SOD		DUT-OMRON	
	F-score	MAE	F-score	MAE	F-score	MAE	F-score	MAE	F-score	MAE
PPM	0.875	0.040	0.942	0.039	0.929	0.034	0.866	0.107	0.801	**0.055**
ASPP	0.876	0.040	0.941	0.040	0.928	**0.033**	0.863	0.110	0.799	**0.055**
PSAM	**0.879**	**0.040**	**0.944**	**0.038**	**0.931**	0.034	**0.874**	**0.104**	**0.813**	0.056

Table 3.3: Quantitative results with F-score and MAE on five challenging datasets: DUTS-TE, ECSSD, HKU-IS, SOD and DUT-OMRON for the different pyramid modules.

In this paper, we propose a novel end-to-end salient object detection method. Con-

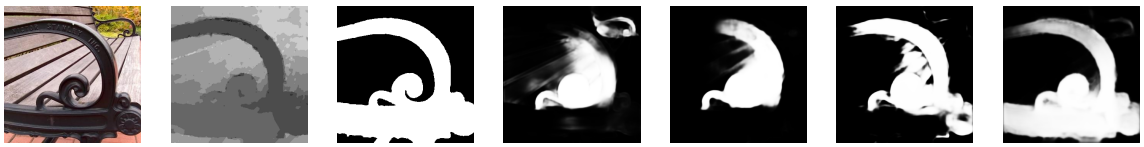
sidering the intrinsic problems of the FPN architecture, a pyramid self-attention module (PSAM) is designed. This module contains different self-attention layers at multiple scales, leading in efficiently capturing multi-scale, high-level features, producing a model which focuses on the high-level semantic information and further enlarges the receptive field. Furthermore, we employ channel-wise attention in lateral connections to reduce the feature redundancy and refine prediction results. Experimental results on five challenging datasets demonstrate that our proposed model surpasses 12 state-of-the-art methods. The ablation experiments further demonstrate the effectiveness of the two proposed modules.

Chapter 4

Progressive Multi-scale Fusion Network for RGB-D SOD

4.1 Introduction

Depth maps can provide effective spatial information for salient object detection in complex scenes. Depth maps were first introduced in [7] to provide complementary information to further enhance the detection performance. In [7] the depth information is provided by the Kinect camera. The importance of depth information is illustrated in a representative example depicted in Figure 4.1. It is seen that RGB-only based method [1] exhibits unsatisfactory prediction results in complex scenes. Compared to Figure 4.1(d) and (e), both (f) and (g) show more detail in the detection saliency map due to the additional depth information. However,



(a) Image (b) Depth (c) GT (d) RGB1 (e) RGB2 (f) RGBD (g) Ours

Figure 4.1: Qualitative visual results of RGB and RGB-D based methods. (a) original image, (b) depth image, (c) the ground truth, (d) PoolNet [1],(d) PSAMNet [2] (Chapter 3), (f) CPFP [3], (g) our proposed method.

erroneous depth information may cause a negative impact on SOD. [3] proposes a cross-modal fusion strategy to overcome this problem by enhancing the contrast between foreground and background objects in depth images. As shown in Figure 4.1(e), in the joint method proposed in [3], more detail is detected in the saliency map compared to RGB-only based method. However, the salient object is still incomplete and non-salient parts are also shown in the saliency map. This result indicates that the quality of depth images is uncertain and depends on the sensor. This observation implies that when we combine image data obtained from visual cameras with depth sensor data, the latter does not necessarily contribute positively to the final outcome of the particular algorithm, due to the possible low quality of the depth sensor data.

Recent works [80, 81, 82, 83, 10] adopt an early fusion strategy, which indicates that the features from different modalities are combined in shallow layers. However, using solely single stream structure cannot generate satisfactory prediction results as demonstrated in column 3 of Figure 4.2. The additional depth information should be fed into the network only if it results in further boosting the detection performance. D3Net [33] adopts multiple streams to extract cross-domain features and fuse them together. Even with the help of extra streams, the same issues also appear as shown in column 4 of Figure 4.2. More, specifically, even with a clear depth map in the top row, it is still difficult to correctly introduce the complementary features from the depth domain. Second, when the non-salient object has low contrast with the salient object in the depth map, such as the bud near the flower in the bottom row, existing methods are misled by the distorted depth map.

To tackle the above issues, we seek alternative fusion strategies. We consider the RGB-Depth feature fusion in a progressive way and propose a novel network with a Mask-Guided Feature Aggregation (MGFA) module, which can take advantage of multi-scale features and effectively introduce useful depth information.

However, multi-scale fusion suffers from the following drawbacks. Directly introducing redundant information can impede performance. The redundant features will lead

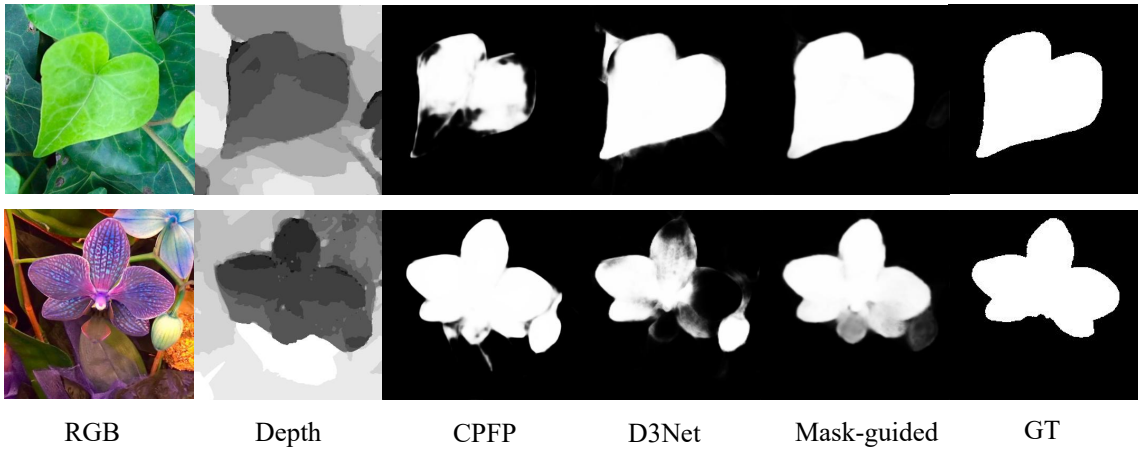


Figure 4.2: Unstatisfactory predictions of existing methods.

to the emergence of objects unrelated to the target object in the final predictions, which could greatly affect the accuracy of the output saliency map. [84] investigates the multi-scale issue and aims at gaining benefits from multi-scale features. More, specifically, the authors in [84] design a transformation-interaction-fusion strategy and propose an aggregate interaction module to utilize the features from adjacent layers, which could introduce less noise due to small sampling rates. However, this method utilizes extra parameters on mutual learning and self-interaction. Considering the same issues, we design a novel refinement module to filter the noise and irrelevant information and further complement semantic information in an effective way.

Different from the regular middle fusion and late fusion [85], we exploit the advantages of both fusion strategies in the overall architecture. Concretely, we use two backbones to extract features separately and aggregate cross-domain features at multi-scales from deep to shallow. In addition, the proposed methods subtly follow the coarse-to-fine pathway to generate better predictions in a progressive way.

Our main contributions can be concluded as follows:

- We construct a two-stream network with a mask-guided feature aggregation module to extract RGB features and depth features. The proposed network considers merging the two-modal features from deep layers to shallow layers in a progressive way and the mask-guided feature aggregation module can

effectively merge two-modal features by the guidance of deep-layer features, leading to the alleviation of the negative impact of unstable depth maps.

- We further design a mask-guided refinement module to diminish the noise features in the multi-scale setting, semantic features can be maintained well due to introducing deep-layer features and the noise can be filtered out efficiently with the guide of the prediction mask.

- We evaluate our proposed method with 11 state-of-the-art SOD methods on five benchmarks. It achieves competitive performance against previous methods with respect to different evaluation metrics.

4.2 Related Work

4.2.1 RGB-D Salient Object Detection

Over the past few years, traditional models [86, 87, 88] utilize jointly the useful attributes from RGB and depth cues to design handcrafted features. However, the use of such type of features limits the performance of traditional methods. More specifically, handcrafted features are normally inconsistent and have low representation ability, leading to unsatisfactory detection results, especially in complex scenarios.

In recent years, deep learning models are adopted to fuse RGB and depth data since fully convolutional networks show superiority on SOD due to their ability in extracting robust features. [3] utilizes contrast priors to enhance the contrast between foreground and background objects in depth maps. [10] designs a single stream network and adopts both early fusion and middle fusion strategies. This method learns the cross-modal interaction in the encoder and further enhances its effect in the decoder. [89] proposes a dense connected structure to integrate cross-modal features, which can produce region-aware dynamic filters to guide the decoder. [90] treats depth images as mid-level or high-level feature maps and adopts

a lightweight network to obtain depth features. Besides, the authors in [90] propose an alternate refinement strategy in a coarse-to-fine manner to alternately use RGB and depth features. [91] investigates adaptive and attention distillation schemes and connects two modalities by designing a depth distiller, which can transfer reliable and avoid erroneous depth information. Different from the aforementioned methods, our work aims to utilize a mask-guided mechanism to alleviate the impact of noise which is caused by low-quality depth features and multi-scale information.

4.2.2 Mask-guided Mechanisms

[92] designs a context-guided cross-modality fusion module to generate attention maps for both RGB and T modalities and then combine them as guidance for feature fusion. [93] proposes a mask-attention module by using the QKV attention mechanism in generative adversarial networks. Unlike these methods, [94] proposes an effective mask-guided attention branch for pedestrian detection. Inspired by this attention mechanism, we propose a simple mask-guided method to filter the cross-domain noise. We take full advantage of the top-down pathway from deep to shallow layers and use the finest predictions at the current stage to generate a mask map, which can effectively mitigate the negative impact of low-quality depth maps.

4.3 The Proposed Framework

In this section, we first introduce the proposed overall architecture and then present two main components in detail, namely, the proposed novel Mask-Guided Feature Aggregation module (MGFA) and the Mask-Guided Refinement Module (MGRM).

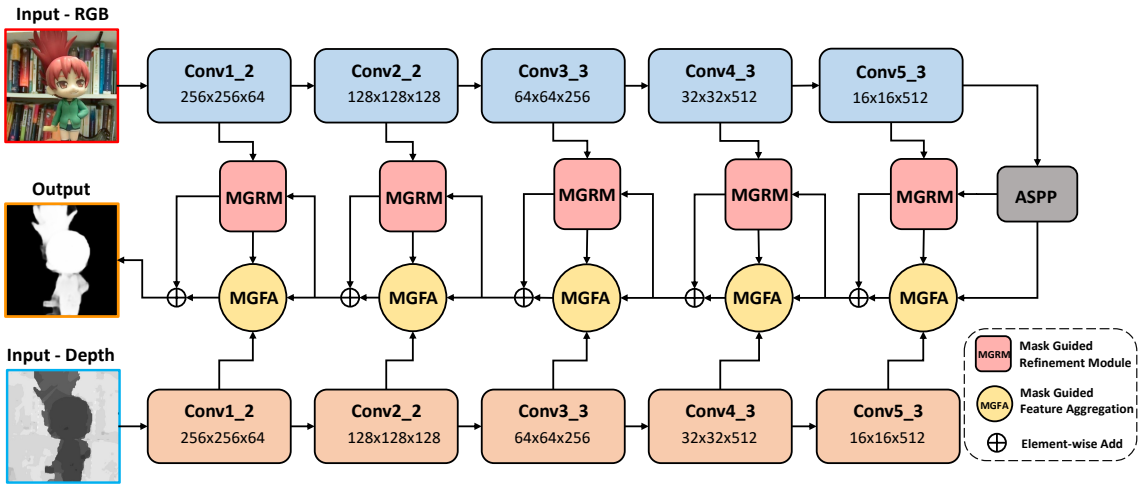


Figure 4.3: The overall structure of our proposed network, where RGB and depth images are fed into two VGG19 backbones and aggregated progressively from deep to shallow layers. ASPP is the atrous spatial pyramid pooling module[4].

4.3.1 The Overall Architecture

We propose a two-stream network which is illustrated in Figure 4.3. This structure follows a coarse-to-fine network structure and has two inputs, namely, RGB images and depth images, which are fed into RGB stream and depth stream respectively. The two streams have the same feature extraction structure. Both of them utilize the 19-layer deep convolutional neural network VGG19 [6] as a backbone to generate multi-level features with 5 different resolutions, which can be denoted as $\{f_r^1, f_r^2, f_r^3, f_r^4, f_r^5\}$ and $\{f_d^1, f_d^2, f_d^3, f_d^4, f_d^5\}$. More specifically, instead of directly combining them, we input an initial prediction map which is generated by a semantic segmentation module that follows the Atrous Spatial Pyramid Pooling (ASPP) approach [4] with RGB and depth features into our proposed MGFA to aggregate two-modal features in the corresponding scales. Then, the aggregated output is fed into another MGFA which treats the output as a saliency map. Furthermore, an MGRM is applied to refine the RGB features by inputting the same saliency map in each scale.

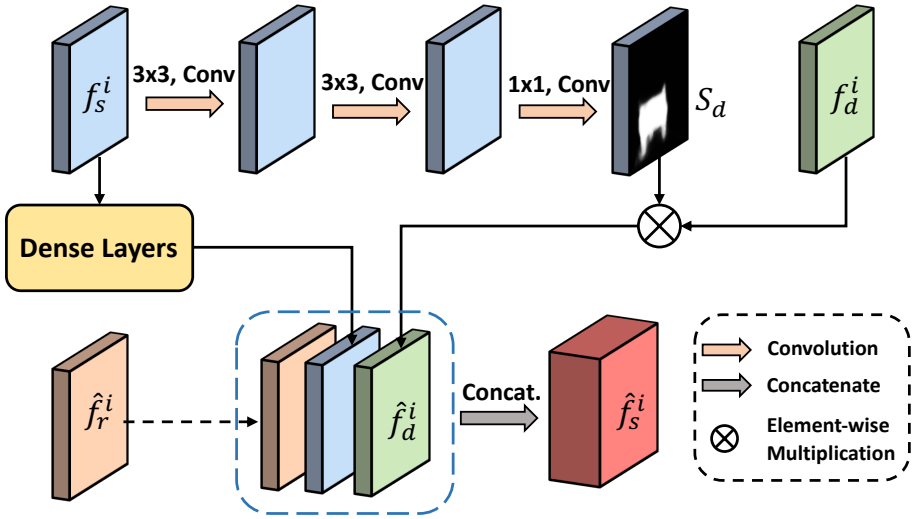


Figure 4.4: The proposed mask-guided feature aggregation. $\hat{f}_r^i$ and f_d^i represent the i-th level of RGB and depth features, respectively. f_s^i is the output of previous MGFA $\hat{f}_s^{i+1}$.

4.3.2 Mask-Guided Feature Aggregation

As mentioned above, the early fusion approach could potentially damage the robust RGB features due to low-quality depth maps and the detection performance also depends on the quality of depth maps. In order to tackle these issues, we propose, as mentioned, a novel Mask-Guided Feature Aggregation (MGFA) module which aims to take full merit of both RGB and depth features. As shown in Figure 4.4, the initial input saliency map of MGFA is generated by the final stage of RGB stream which is termed as f_s. Inspired by [94], the mask S_d is obtained as follows:

$$S_d = \sigma(N_1(f_s)) \tag{4.1}$$

where N_1 indicates two 3x3 convolution layers and a 1x1 convolution layer. σ refers to the sigmoid function. Then, the filtered depth feature map denoted by $\hat{f}_d$ is obtained by applying element-wise product between S_d and f_d as follows:

$$\hat{f}_d = S_d \cdot f_d \tag{4.2}$$

After processed by the mask-guided operation, the unreliable features from low-quality depth images are filtered and then we use a dense layer [16] to obtain a more powerful feature map S_{dense} with abundant receptive fields and information details. The integrated feature map $\hat{f}_s$ is generated by the different modal features, namely, S_{dense}, $\hat{f}_d$ and $\hat{f}_r$, where $\hat{f}_r$ is the refined RGB feature map described in section 3.3.

$$\hat{f}_s = Cat(\hat{f}_r, \hat{f}_d, S_{dense}) \tag{4.3}$$

where Cat represents the concatenation operation. Finally, the output $\hat{f}_s$ is fed into the next MGFA module with $\hat{f}_r$ to guide the shallower layer's depth information and the ground truth is applied to further supervise the i-th output.

This progressive feature aggregation strategy has the following benefits. First, transferring two-modal information into multi-scale features and aggregating them progressively from deep to shallow can overcome the drawbacks. Specifically, the generated mask can effectively filter the noise from depth features, especially within shallow levels, leading to alleviating the impact of low-quality depth images. Furthermore, along with the RGB and depth information, the output from the previous deeper level is provided to further refine the feature aggregation and complement the semantic features.

4.3.3 Mask-Guided Refinement Module

Multi-scale fusion strategy combines complementary information and global context information from both RGB and depth modalities. However, it also introduces irrelevant information which impedes the performance of salient object detection. As shown in Figure 4.1(d) and (e), not only the salient object is detected, some irrelevant parts are also presented. To tackle this problem, we propose a Mask-Guided Refinement Module(MGRM) to filter the noise and irrelevant information in multiple scales. As illustrated in Figure 4.5, sharing the similar mask-guided idea to MGFA,

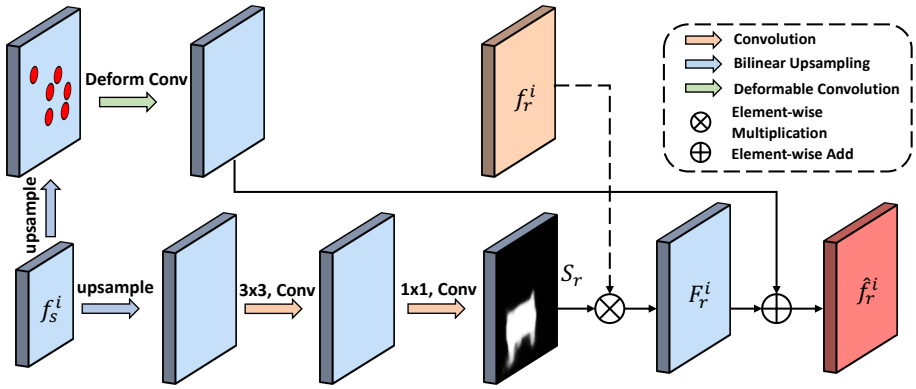


Figure 4.5: The proposed mask-guided refinement module.

the RGB feature maps can be refined by the deeper-layer feature maps:

$$S_r = \sigma(N_2(f_s)) \tag{4.4}$$

$$F_r = S_r \cdot f_r \tag{4.5}$$

where N_2 indicates a 3x3 convolution layer and a 1x1 convolution layer. S_r indicates the mask for RGB features and F_r represents the filtered RGB features. MGRM also diversifies the receptive fields of multiple scales. Specifically, the receptive field of RGB features in each scale can be enlarged by introducing the features from deeper layers:

$$\hat{f}_r = deformable(Up(f_s, f_r)) + F_r \tag{4.6}$$

where Up represents bilinear interpolation operation to resize the deeper-layer feature map f_s to the same size with f_r. $deformable$ refers to a deformable convolution layer with a flexible kernel to extract f_s after upsampling. Finally, the filtered RGB F_r is added to generate the output $\hat{f}_r$.

Figure 4.6 shows the effectiveness of MGRM on different scales. The first row refers to the four different scales' outputs without the refinement of MGRM and the bottom row represents the corresponding outputs with the MGRM. Obviously, the drawing board displayed in the background misleads the model to detect this

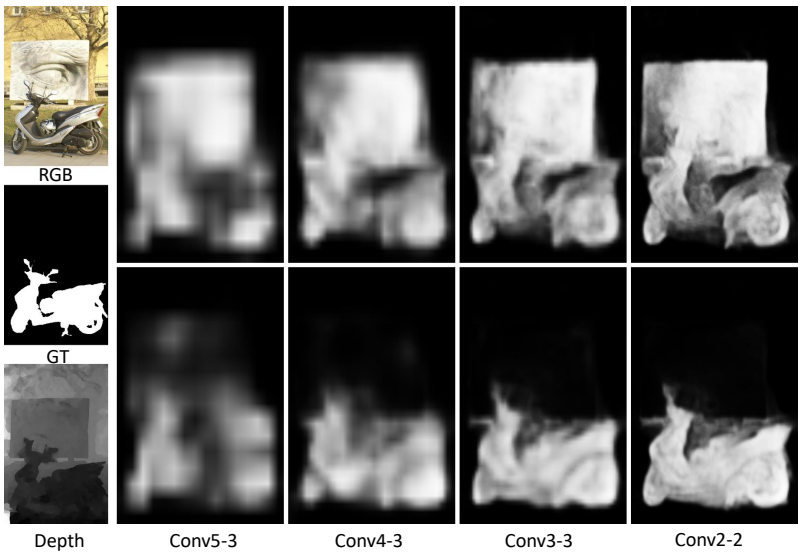


Figure 4.6: Predictions from four different levels of the proposed network. $Conv5-3$ represents the saliency map which is generated by f_r^5 and f_d^5. Saliency maps in the first row are generated solely by MGFA whereas the bottom row maps are produced by MGFA and MGRM.

non-salient object due to the low contrast of the RGB map. Without the MGRM, noise is also introduced with the global context information during multi-scale fusion. Compared to the erroneous phenomena described above, it is observed that MGRM can effectively introduce semantic features and filter the irrelevant features, leading to further detailed refinement of the saliency map. The proposed progressive fusion strategy is illustrated in algorithm 1

4.4 Experiments

4.4.1 Datasets

We conduct our experiments on five publicly challenging RGB-D datasets. NJUD [29] contains 1,985 stereo image pairs, which are captured by a Fuji stereo camera from the internet, 3D movies and photographs. NLPR [30] contains 1,000 image pairs, including outdoor and indoor locations which are captured by a standard Microsoft Kinect. LFSD [31] and RGBD135 [32] include 100 light fields images and 135 images, respectively. SIP [33] is a large-scale dataset that includes 929 high-resolution images.

Algorithm 1 Progressive Multi-scale Fusion.

Input: f_r^i is the i-th level RGB features, f_d^i is the i-th level depth features, f_s^i is the deep semantic features, the initial f_s^5 is generated by the ASPP.

for i = 5, 4, 3, 2, 1 **do**
  $S_r^i = \text{Mask}(f_s^i)$ ▷ Stage 1;
  $F_r^i = S_r^i \cdot f_r^i$;
  $\hat{f}_r^i = deformable(Up(f_s^i, f_r^i)) + F_r^i$;
  $S_d^i = \text{Mask}(f_s^i)$ ▷ Stage 2;
  $\hat{f}_d^i = S_d^i \cdot f_d^i$;
  $S_{dense}^i = \text{Dense}(f_s^i)$;
  $\hat{f}_s^i = Cat(\hat{f}_r^i, \hat{f}_d^i, S_{dense}^i)$;
  $Output^i = \hat{f}_s^i + \hat{f}_r^i$ ▷ Stage 3;
  if i != 1 **then**
    $f_s^{i-1} = Output^i$;
  end if
end for
return $Output^1$;

This dataset captures multiple persons in diverse scenes by a real smartphone. We adopt the same training dataset as in [34]. This composited dataset contains 800 samples from DUT-RGBD [34], 1,485 samples from NJUD and 700 samples from NLPR. The data that support the findings of this study are available in [34], [29], [30], [31], [32] and [33].

4.4.2 Evaluation Metrics

We adopt five widely used evaluation metrics that can represent the quality of models, namely, the Precision-Recall (PR) Curve, the F-measure score (2.37), the Mean Absolute Error (2.39), the S-measure (2.40) and the E-measure (2.49).

4.4.3 Implementation Details

Our model is implemented using Pytorch Toolbox and trained on a P100 GPU for 40 epochs with mini-batch size 4. We use two identical VGG19 as feature extraction backbones for RGB stream and depth stream. Both RGB and depth images are resized to 256×256. To avoid overfitting, simple flipping and rotating are adopted

to augment the training dataset. The initial learning rate is set to 1e-3 and using poly policy [95]. We adopt a 0.0005 weight decay for the stochastic gradient descent (SGD) with a momentum of 0.9 and binary cross entropy loss function for supervision. The entire training parameters are suggested by [10].

Table 4.1: Quantitative comparison including the maximum of F_β, S_α, E_θ, and M, over five widely evaluated datasets. ↑ & ↓ represent higher and lower is better, respectively. The best two scores are highlighted in **red** and **blue**.

	Metric	CDCP	DF	CTMF	AFNet	MMCI	TANet	DMRA	CPFP	D3Net	A2dele	DANet	Ours
NJUD	$F_\beta \uparrow$	0.661	0.789	0.857	0.804	0.868	0.888	0.896	0.890	0.903	0.890	**0.910**	**0.919**
	$S_\alpha \uparrow$	0.672	0.735	0.849	0.772	0.859	0.878	0.885	0.878	0.895	0.867	**0.899**	**0.910**
	$E_\theta \uparrow$	0.751	0.818	0.866	0.847	0.882	0.909	0.920	0.900	0.901	0.914	**0.922**	**0.926**
	$M \downarrow$	0.182	0.151	0.085	0.100	0.079	0.061	0.051	0.053	0.051	0.052	**0.045**	**0.045**
NLPR	$F_\beta \uparrow$	0.687	0.752	0.841	0.816	0.841	0.876	0.888	0.884	0.904	0.891	**0.916**	**0.924**
	$S_\alpha \uparrow$	0.724	0.769	0.860	0.799	0.856	0.886	0.898	0.884	0.906	0.889	**0.915**	**0.923**
	$E_\theta \uparrow$	0.786	0.840	0.869	0.884	0.872	0.926	0.942	0.920	0.934	0.937	**0.949**	**0.942**
	$M \downarrow$	0.115	0.110	0.056	0.058	0.059	0.041	0.031	0.038	0.034	0.031	**0.028**	**0.030**
DES	$F_\beta \uparrow$	0.651	0.625	0.865	0.775	0.839	0.853	0.906	0.882	0.917	0.897	**0.928**	**0.933**
	$S_\alpha \uparrow$	0.709	0.685	0.863	0.770	0.848	0.858	0.899	0.872	0.904	0.883	**0.924**	**0.926**
	$E_\theta \uparrow$	0.810	0.806	0.911	0.874	0.904	0.919	0.944	0.927	0.956	0.918	**0.968**	**0.964**
	$M \downarrow$	0.120	0.131	0.055	0.068	0.065	0.046	0.030	0.038	0.030	0.030	**0.023**	**0.027**
LFSD	$F_\beta \uparrow$	0.680	0.854	0.815	0.780	0.813	0.827	0.872	0.850	0.849	0.858	**0.871**	**0.894**
	$S_\alpha \uparrow$	0.658	0.786	0.796	0.738	0.787	0.801	0.847	0.828	0.832	0.833	**0.849**	**0.874**
	$E_\theta \uparrow$	0.737	0.841	0.851	0.810	0.840	0.851	0.899	0.867	0.860	0.875	**0.881**	**0.907**
	$M \downarrow$	0.199	0.142	0.120	0.133	0.132	0.111	0.076	0.088	0.099	0.077	**0.079**	**0.072**
SIP	$F_\beta \uparrow$	0.544	0.704	0.720	0.756	0.840	0.851	0.847	0.870	0.882	0.855	**0.892**	**0.913**
	$S_\alpha \uparrow$	0.595	0.653	0.716	0.720	0.833	0.835	0.800	0.850	0.864	0.828	**0.875**	**0.896**
	$E_\theta \uparrow$	0.722	0.794	0.824	0.815	0.886	0.894	0.858	0.899	0.903	0.890	**0.915**	**0.923**
	$M \downarrow$	0.224	0.185	0.139	0.118	0.086	0.075	0.088	0.064	0.063	0.070	**0.054**	**0.051**

4.4.4 Comparison with State-of-the-art Results

We compare our model with 11 state-of-the-art methods on five challenging benchmark datasets, including CDCP [96], DF [97], AFNet [98], CPFP [3], MMCI [99], TANet [100], CTMF [101], D3Net [33], DMRA [34], A2delde [91] and DANet [10]. For fair comparisons, we directly use the saliency maps released by the authors or the evaluation results pre-computed by the authors.

Quantitative Evaluation. Table 4.1 and Figure 4.7 show the quantitative results in comparison to 11 state-of-the-art methods in terms of four evaluation metrics on five challenging datasets. More specifically, it is observed in Table 4.1 that our proposed method outperforms all other methods in terms of maximum F_β and S_α across five datasets. Especially, our method outperforms all other methods by a

large margin in four metrics on SIP, which is a large-scale dataset with relatively complicated scenes. Figure 4.7 shows the comparison results on PR curve, which computes the precision and recall between the binary mask and ground truth. It can be clearly observed that our method which is represented by the red line outperforms other state-of-the-art methods. This observation illustrates the better performance of our model at different thresholds.

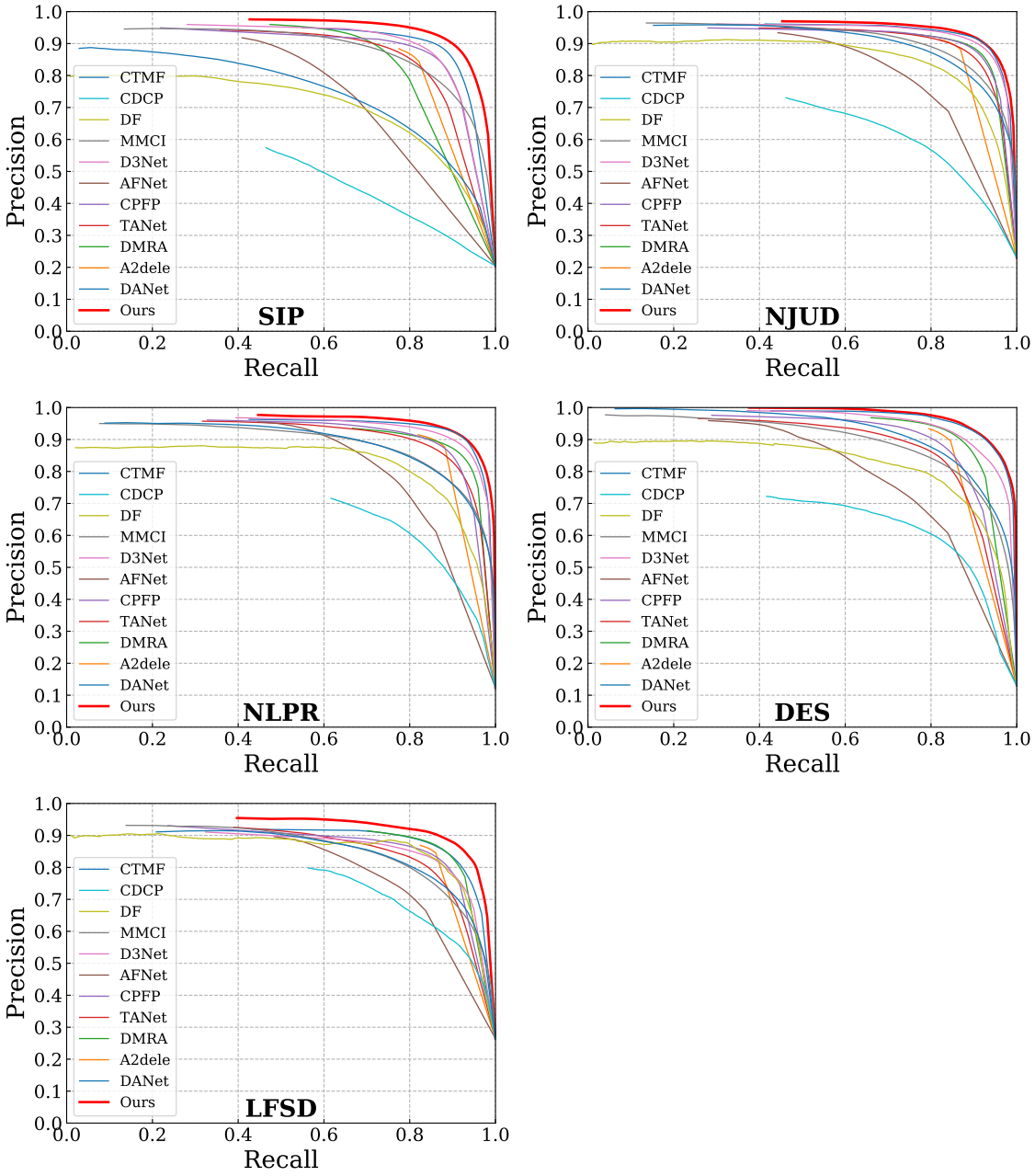


Figure 4.7: Results with PR curves on five benchmark datasets: DUTS-TE, ECSSD, HKU-IS, SOD and DUT-OMRON. x-axis represents the recall rate and y-axis represents the precision.

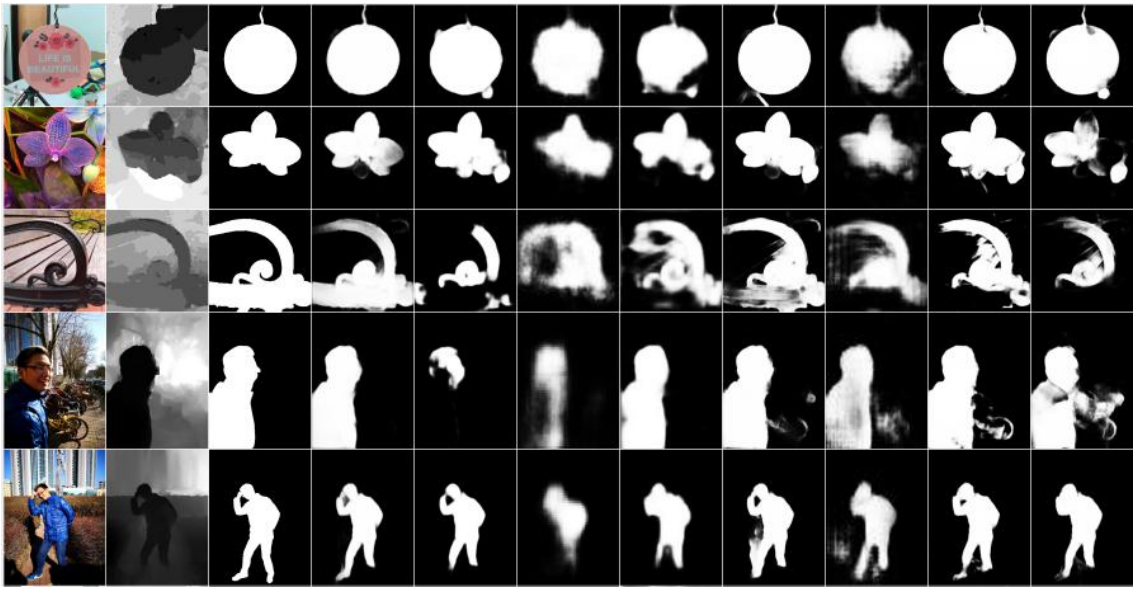


Figure 4.8: Visual comparisons with state-of-the-arts in different scenarios, including complicated background, low-quality depth image, low contrast between foreground and background in RGB and depth and misleading objects.

Qualitative Evaluation. Figure 4.8 illustrates the visual comparison with other state-of-the-art methods. Generally, the results from our model are more similar to the ground truth in different scenarios, which shows the superiority of our method. For instance, in low contrast scenes (row 3 and row 5), our method can accurately detect the salient object with more semantic details. In addition, in complex environments, even the depth maps have high contrast between the foreground and background (row 1, row 2 and row 5), the existing methods still detect undesired parts due to the complicated background in the RGB maps. Compared to other methods, our method can effectively filter the non-salient parts and focus more on the salient objects in complex environments. Furthermore, our method shows an accurate prediction in a challenging situation (Figure 4.6).

4.4.5 Ablation Studies

In this section, we conduct a series of experiments to verify the effectiveness of the proposed methods. The baseline is an FPN network that is built on a VGG19

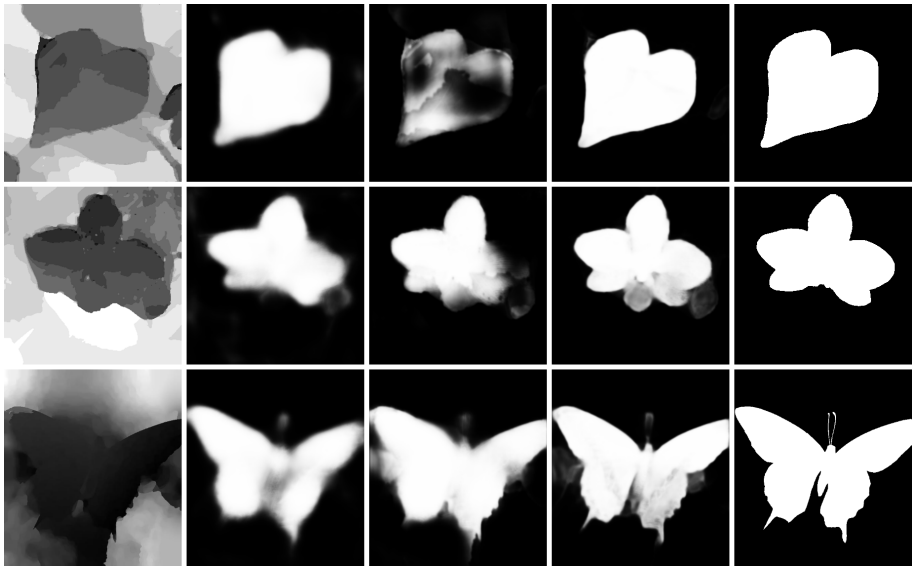


Figure 4.9: Effectiveness of MGFA. *Base* represents the FPN baseline, *Add* means combining two-modal features by summation in the same two-stream framework.

backbone.

Mask-Guided Feature Aggregation. Figure 4.9 illustrates the effectiveness of MGFA. *Add* and $MGFA$ represent two multi-scale fusion strategies in the same two-stream network. It can be observed in row 1 and row 2 that directly adding the RGB and depth features can impede the detection performance compared to the results of *Base*. The bottom row shows the prediction results with a low-contrast depth map. Although simple addition between two-modal features can improve the semantic features, irrelevant information is also introduced. Compared to direct combination in the multi-scale fusion framework, MGFA shows better performance on cross-modal feature aggregation.

Additionally, Table 4.2 shows the quantitative results of MGFA in the second column. It is demonstrated that the MGFA improves the baseline in all metrics across three datasets. Especially, the MGFA boosts the performance by a large margin on SIP, providing gains of 8.9%, 7.2%, 5.6% and 4.7% in terms of the maximum F_β, S_α, E_θ and M respectively. On the other hand, Figure 4.10 illustrates the visual results after adopting MGFA in the fourth column. More specifically, row 1 shows a salient object in a particular scene where the foreground and background possess similar

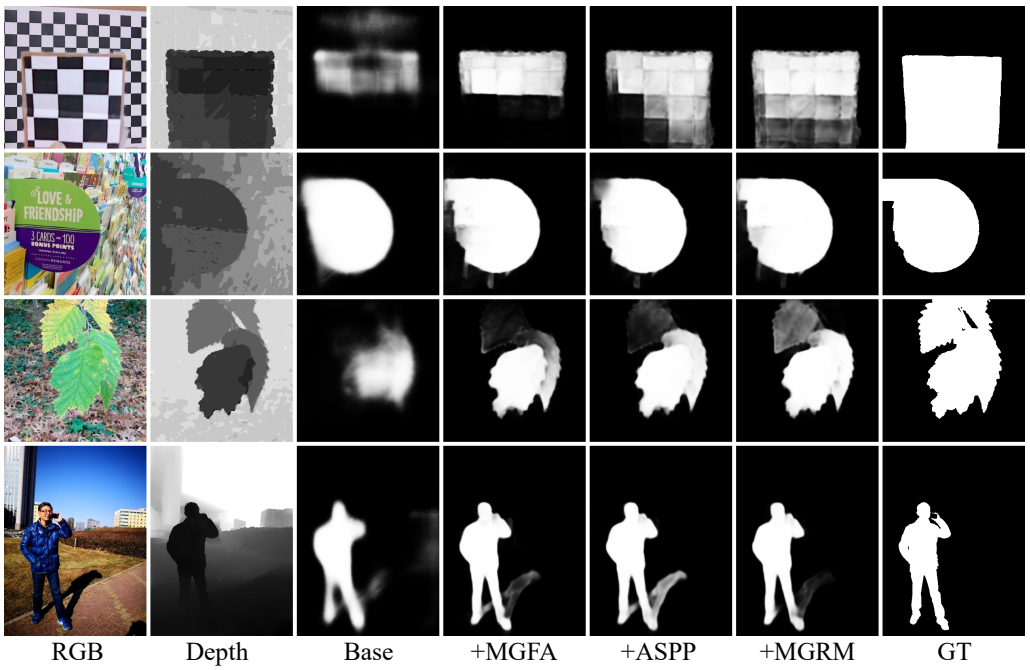


Figure 4.10: Visual comparisons of the proposed modules. For each experiment we employ the corresponding modules on the framework. Especially, MGFA and MGRM are employed on five scales and the ASPP module is only applied at the end of the RGB stream.

structures. Row 2 and row 3 show complex background in RGB maps. Row 4 shows a depth map in the case of low contrast between the salient object under consideration and the background. Compared to the predictions of baseline, using MGFA could alleviate the impact of low-quality depth maps and effectively exploit their semantic information. Furthermore, due to the smaller effective receptive field [102], we add an ASPP module at the end of RGB stream to further enlarge the receptive field for large-scale objects.

Mask-Guided Refinement Module. We employ the MGRM to build the proposed model. The effectiveness of MGRM can be demonstrated in the last column of Table 4.2. It can be observed that all evaluation metrics are improved by the proposed MGRM. Evidently, this module improves the overall performance and reduces the error by a large margin on LFSD, providing gains of 1.6%, 1.1%, 1.3% and 0.9% in terms of the maximum F_β, S_α, E_θ and M respectively. Furthermore, Figure 4.10 illustrates the effectiveness of MGRM in visual maps in the sixth column. Compared to the previous columns, it is obvious that row 1 and row 3 in the sixth column

Table 4.2: Ablation analyses on NLPR, LFSD and SIP. *Base* means an FPN network built on a VGG19 backbone. MGFA and MGRM are two proposed modules. ASPP is an Atrous Spatial Pyramid Pooling module.

	Metric	Base	+MGFA	+ASPP	+MGRM
NLPR	$F_\beta \uparrow$	0.865	0.917	0.920	**0.924**
	$S_\alpha \uparrow$	0.877	0.917	0.922	**0.923**
	$E_\theta \uparrow$	0.893	0.934	0.940	**0.942**
	$M \downarrow$	0.051	0.033	0.031	**0.030**
LFSD	$F_\beta \uparrow$	0.856	0.865	0.878	**0.894**
	$S_\alpha \uparrow$	0.825	0.850	0.863	**0.874**
	$E_\theta \uparrow$	0.858	0.894	0.894	**0.907**
	$M \downarrow$	0.112	0.082	0.081	**0.072**
SIP	$F_\beta \uparrow$	0.821	0.910	0.911	**0.913**
	$S_\alpha \uparrow$	0.818	0.890	0.891	**0.896**
	$E_\theta \uparrow$	0.862	0.918	0.920	**0.923**
	$M \downarrow$	0.101	0.054	0.055	**0.051**

Table 4.3: Effectiveness of mask-guided mechanism on NLPR, LFSD. *Sum* and *Concat* indicate that we fuse RGB and depth features by employing summation and concatenation operation, respectively.

	Metric	Base	Sum	Concat	MGFA
NLPR	$F_\beta \uparrow$	0.865	0.885	0.884	**0.917**
	$S_\alpha \uparrow$	0.877	0.886	0.889	**0.917**
	$E_\theta \uparrow$	0.893	0.912	0.914	**0.934**
	$M \downarrow$	0.051	0.044	0.043	**0.033**
LFSD	$F_\beta \uparrow$	0.856	0.850	0.853	**0.865**
	$S_\alpha \uparrow$	0.825	0.819	0.824	**0.850**
	$E_\theta \uparrow$	0.858	0.869	0.876	**0.894**
	$M \downarrow$	0.112	0.099	0.096	**0.082**

are closer to the ground truth, which demonstrates that the MGRM can effectively complement high-level semantic features. On the other hand, row 2 and row 4 show a better performance in preserving details, especially in the bottom area where the contrast is quite low in the depth maps. These results indicate that the MGRM can further refine the saliency maps.

Effectiveness of mask-guided mechanism. To further demonstrate the effectiveness of the proposed mask-guided mechanism, we employ different features fusion methods on the two-stream structure. Experimental results in Table 4.3 show that simple summation and concatenation are suboptimal fusion strategies due to the introduction of both depth information and noise. It is worth noting that both *Sum*

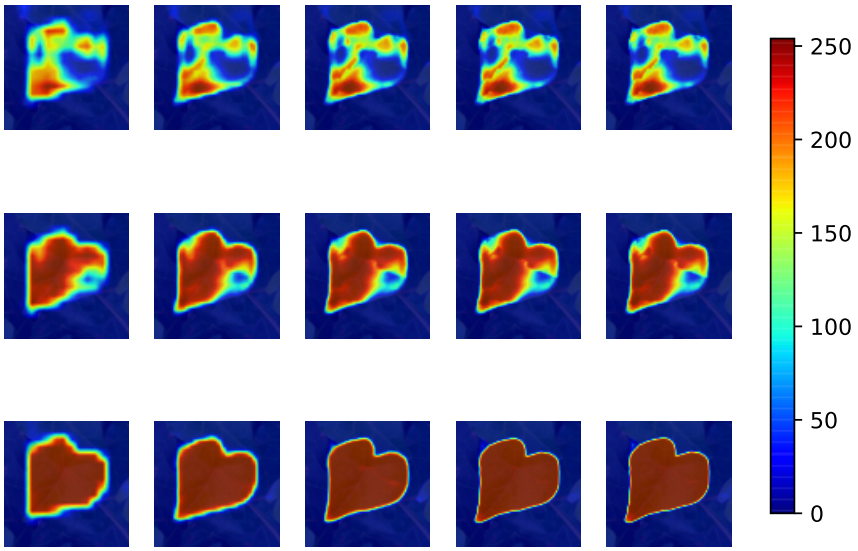


Figure 4.11: Effectiveness of mask-guided mechanism in RGB-D. Three rows from top to bottom indicate different fusion methods, namely, summation, concatenation and mask-guided. Columns from left to right represent five stages from deep to shallow layers.

and *Concate* reduce the detection accuracy on LFSD in F_β and S_α due to the noise issue caused by low quality depths. Specific examples have been shown in Figure 4.11. It is clearly observed that compared to other cross-modal fusions, our mask-guided mechanism can effectively introduce the depth features and progressively refine the saliency map from deep to shallow layers.

Hyperparameters Setting. We conduct experiments on a different number of scales to demonstrate the proposed progressive multi-scale fusion strategy. Table 4.4 shows the results on three different scales across three datasets. Specifically, s represents the number of scales. For instance, $s = 3$ indicates that we only combine the $\{f_r^3, f_r^4, f_r^5\}$ and $\{f_d^3, f_d^4, f_d^5\}$ by employing three MGFAs and MGRMs. It is seen that using only three scales can obtain a large gain on SIP in terms of all metrics, namely, 8.1% on F_β, 6.7% on S_α, 5.5% on E_θ and 4.4% on M. We have shown that the semantic features are mainly contained in the deep layers, therefore, three-scale

Table 4.4: Quantitative comparison with different scales on NLPR, LFSD and SIP.

	Metric	Base	s=3	s=4	s=5
NLPR	$F_\beta \uparrow$	0.865	0.907	0.921	**0.924**
	$S_\alpha \uparrow$	0.877	0.914	0.921	**0.923**
	$E_\theta \uparrow$	0.893	0.935	0.941	**0.942**
	$M \downarrow$	0.051	0.033	0.030	**0.030**
LFSD	$F_\beta \uparrow$	0.856	0.891	0.891	**0.894**
	$S_\alpha \uparrow$	0.825	0.865	0.872	**0.874**
	$E_\theta \uparrow$	0.858	0.895	0.901	**0.907**
	$M \downarrow$	0.112	0.079	0.076	**0.072**
SIP	$F_\beta \uparrow$	0.821	0.902	0.907	**0.913**
	$S_\alpha \uparrow$	0.818	0.885	0.890	**0.896**
	$E_\theta \uparrow$	0.862	0.917	0.918	**0.923**
	$M \downarrow$	0.101	0.057	0.055	**0.051**

fusion can achieve accurate performance. Furthermore, the experimental results also indicate that increasing the number of scales could lead to better overall performance. In our experiment, s is set to 5 for the progressive multi-scale fusion.

4.4.6 Discussion

There are still some drawbacks of the proposed model. Firstly, the proposed mask-guided methods only mitigate the noise issue in low-quality depth and multi-scale feature maps. If the initial mask is not accurate, the final prediction may still have false positives. Secondly, in some easy cases, where the RGB and depth maps are easy to extract features and generate satisfactory predictions at multiple scales, this mask-guided mechanism may provide a minor improvement compared to simple summation and concatenation. Furthermore, due to the utilization of two-stream architecture, the final model size is larger than single stream methods.

4.5 Summary

In this paper, we aim to improve the prediction accuracy of RGB-D based SOD. We design a progressive multi-scale fusion architecture that is built on a two-stream framework. We propose a novel cross-modal feature aggregation module that could

effectively introduce the depth information and alleviate the noise from low-quality depth maps. Furthermore, we propose a mask-guided refinement module to refine the salient maps and improve performance. This module could complement high-level semantic features and filter the noise from multi-scale features due to the progressive mask-guided strategy. Both quantitative and qualitative experimental results demonstrate the effectiveness of the proposed modules verifying that the proposed framework achieves competitive results on five public challenging benchmarks.

Chapter 5

Dynamic Knowledge Distillation for RGB-D SOD

5.1 Introduction

Due to the complicated real-life scenarios, RGB-based SOD still fails in generating satisfactory prediction maps. In order to overcome this issue and obtain better detection performance in complex scenarios, depth images along with an independent network have been introduced to provide supplementary information. Specifically, Figure 5.1 illustrates three fusion methods in RGB-D based SOD. Existing state-of-the-art methods mainly adopt late fusion or multi-scale fusion and focus on designing feature-enhanced modules and complicated feature-fusion modules, which indeed improve the overall detection results. However, due to the processing of a high volume of information, the models tend to become extremely complicated, an issue that weakens the practicality of SOD using RGB-D data. The early fusion in Figure 5.1(c) integrates the separate inputs into a unified representation before the feature extraction process. It provides an alternative strategy to lighten the model but suffers from the noise issue caused by low-quality depth information. That motivates us to seek to explore the potential of early fusion and compress the model

size for SOD while maintaining high detection accuracy.

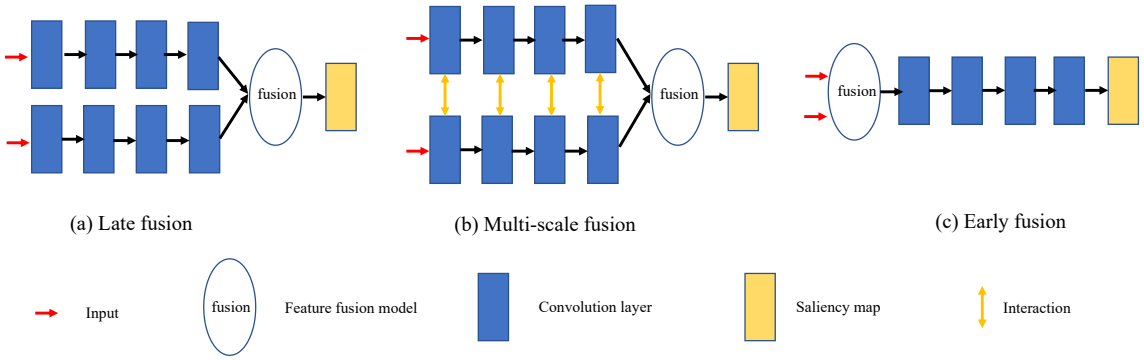
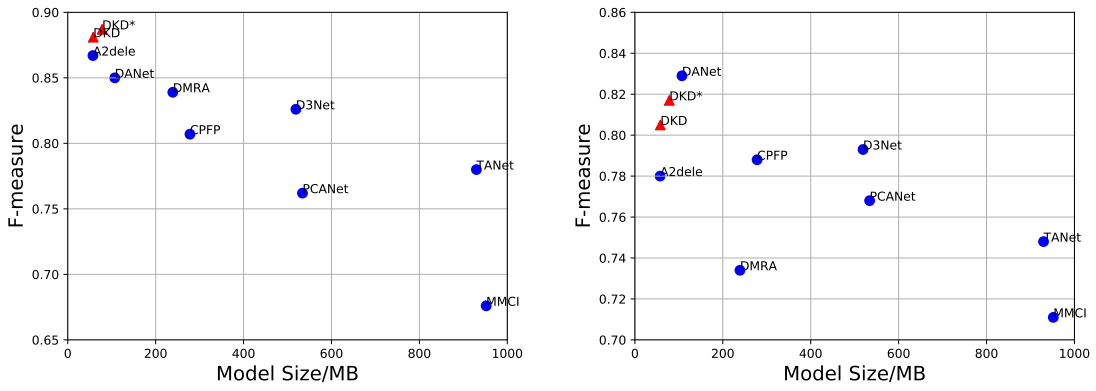


Figure 5.1: Comparison of three fusion strategies.

Recently, Knowledge Distillation (KD) has been proposed [103] and the main idea is that a student model mimics a cumbersome model, namely a so-called teacher model, to achieve competitive performance. In other words, a lightweight network can reach a similar performance to a cumbersome network by KD without increasing the number of parameters. Similar to how humans operate, this teacher-student learning process can be implemented in a simple and effective way, which forces the student model to directly learn the final prediction of the teacher model.



(a) NLPR (b) SIP

Figure 5.2: Weighted F-measure and model size on NLPR and SIP datasets. Our lightweight model with simple early fusion can achieve satisfactory detection results over different datasets. DKD and DKD* refer to the proposed dynamic knowledge distillation with VGG16 and VGG19 backbones respectively.

KD has been applied in a range of machine learning applications. Zhang [104] utilizes the knowledge distillation to RGB saliency detection and proposes an efficient model by reducing the number of channels. Piao [91] explores the cross-modal distillation

on RGB-D data and uses an adaptive weight to distill the depth knowledge from the teacher model. Nevertheless, both adjust their student networks according to the teacher networks and distillation strategies. Besides, the adaptive distillation [91] is proposed for the cross-modal distillation and only considers the performance of the teacher model, which limits the utilization of this KD method.

In order to tackle the above issues from a new perspective, we use a concise framework based on the early fusion strategy for RGB-D based SOD and propose a dynamic knowledge distillation weight to help the model pay more attention on hard samples by considering both teacher and student performance. We also investigate the issue of RGB-D early fusion strategy in distillation and propose a simple noise elimination method to mitigate the impact of distorted training data caused by low-quality depth maps. Combing these two methods can lead to a reasonable distillation strategy for RGB-D saliency detection. Our final model achieves a good balance between accuracy and model size on widely used benchmarks as shown in Figure 5.2. In a nutshell, our main contributions can be summarised as follows:

- We propose a novel dynamic distillation strategy, which can adaptively assign the distillation weight by simultaneously considering the detection performance of the teacher and student networks within the training stage. As a result, the final model can pay more attention to hard samples and improve the overall performance.

- We propose a noise elimination method by taking full merit of knowledge prior from the teacher network to alleviate the impact of depth maps in low quality. The student network can take benefit from this method without increasing extra parameters and computations.

- We adopt a single stream for RGB-D SOD in order to bypass the depth network and avoid designing a complicated model. This single stream achieves competitive performance by only using VGG16 (57.9MB) and VGG19 (78.2MB), which are more applicable for practical use. Extensive experimental results

on five benchmarks demonstrate that our methods can achieve competing performance within a fast lightweight architecture.

5.2 Knowledge Distillation

Knowledge Distillation was formally publicized by [103] in a teacher-student learning framework. This method proposes an effective way to compress model size and attempts to imitate human beings' learning mechanisms. Cheng [105] designs mathematical metrics to quantify and compare the methods of learning from the teacher model and learning from raw data. They explain the superiority of KD in three aspects. First, more reliable visual concepts can be learned through KD. Second, KD makes the model able to learn various concepts simultaneously. Furthermore, learning from KD can generate more stable optimization directions in the training phase. Recently, KD has been used in SOD tasks. Zhang [104] designs the student model by reducing the number of channels and applies multi-scale KD on the corresponding scales between teacher and student models. Piao [91] applies cross-modal distillation on RGB-D based SOD and proposes an adaptive distiller to distill the depth information, which alleviates the impact of low-quality depth maps. Different from the aforementioned methods, our method takes both the performance of teacher and student models into consideration and generates a dynamic weight to control the regular teacher-student KD process. In addition, we analyze the depth issues in the specific RGB-D SOD task and optimize the training phase through a threshold.

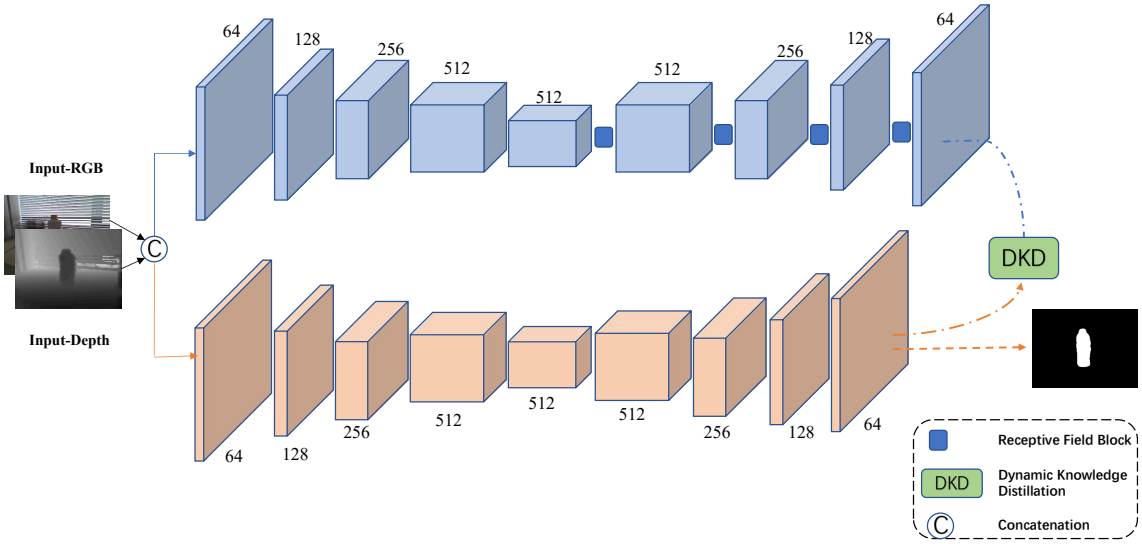


Figure 5.3: Our framework consists of two stages. We adopt a cumbersome network as the teacher model (top) and a feature pyramid network as the student model (bottom).

5.3 Methodologies

5.3.1 Overview

Existing methodologies for RGB-D SOD tend to build two-stream networks in order to process RGB and depth features separately. This two-stream design could improve detection performance but meanwhile introduces a large number of parameters, which increases the complexity and reduces the practicality of models. Feature pyramid network (FPN) [18] is an effective structure that utilizes multi-scale features in different resolutions to achieve detection tasks. In this work, we do not focus on designing networks and only adopt the classic FPN based on a VGG16 and a VGG19 [6] as the student model. In order to obtain a stronger teacher model, we employ four receptive field blocks [106] in multi-scale layers to boost the detection performance. Considering different cross-modal fusion strategies, we choose the simple early fusion way which directly concatenates RGB images and depth images to form four-channel inputs. Similar to normal knowledge distillation, we transfer the probability distribution of the final layer from the teacher model to the student model by utilizing the so-called dynamic KD (Figure 5.3).

5.3.2 Dynamic Knowledge Distillation

As mentioned above, Knowledge Distillation (KD) benefits the student model but the weight of knowledge transfer is still hand-designed. Piao [91] proposes an adaptive weight for cross-modal distillation. However, in [91] they only distill the depth information by considering the performance of the teacher model. In our method, we consider both performances of teacher and student networks and combine these two factors as a dynamic weight for KD.

Concretely, the accuracy of the teacher model represents the detection performance which also indicates the confidence of knowledge. Inspired by IOU used in SOD, we design a dynamic factor α_t to modulate the correct knowledge which can be transferred from the teacher model as follows:

$$\alpha_t = \frac{P_t \cdot G}{P_t + G - P_t \cdot G} \tag{5.1}$$

where P_t and G represent the prediction of the teacher model and the ground truth, respectively. α_t indicates the confidence of knowledge that can be transferred to the student model. Then, we propose another dynamic factor β_s to show the degree of desired knowledge for the student model as follows:

$$\beta_s = 1 - \frac{P_s \cdot G}{P_s + G - P_s \cdot G} \tag{5.2}$$

where P_s represents the prediction of student model. This dynamic factor β_s indicates the knowledge error. In other words, knowledge distillation should also consider the current performance of the student model. β_s is inversely related to the accuracy between the output of the student model and the ground truth. This indicates that hard samples which have large losses need to learn more from the teacher model. Therefore, we propose a simple and effective formulation to find a plausible distillation weight $\theta_{t,s}$:

$$\theta_{t,s} = tanh(\alpha_t{}^p \cdot \beta_s{}^{1-p}) \tag{5.3}$$

here $tanh$ is treated as a scale function:

$$tanh(x) = \frac{exp(x) - exp(-x)}{exp(x) + exp(-x)} \tag{5.4}$$

More specifically, we define the $\theta_{t,s}$ by the weighted geometric mean of the knowledge confidence α_t from the teacher and the knowledge error β_s from the student. We define the hyper-parameter $p \in [0, 1]$ to balance the ratio between the teacher and student networks. It is worth noting that a large variation of $\theta_{t,s}$ leads to convergence issues in the training phase. In this case, we further use a $tanh$ function to scale the $\theta_{t,s}$. The overall loss function can be formulated as:

$$L_{dynamic} = \theta_{t,s} L_{KL}(P_s, P_t) + (1 - \theta_{t,s}) L_{CE}(P_s, G) \tag{5.5}$$

where L_{KL} is the Kullback-Leibler divergence loss and L_{CE} represents the cross-entropy loss. In the final network, we set the distillation temperature to 5 in L_{KL} and $p = 0.7$ according to multiple experiments.

5.3.3 Noises Elimination with the Dynamic Knowledge Distillation

As mentioned above, we simplify the procedure of knowledge distillation and the student network architecture in RGB-D task. Concretely, we only distill the final output distribution and abandon the depth stream by concatenating RGB and depth maps to form a four-channel input. However, this fusion strategy suffers from the noise issue caused by low-quality depth information. As illustrated in Figure 5.4, we investigate the reasons that cause the distortion of depth maps: 1) Besides the

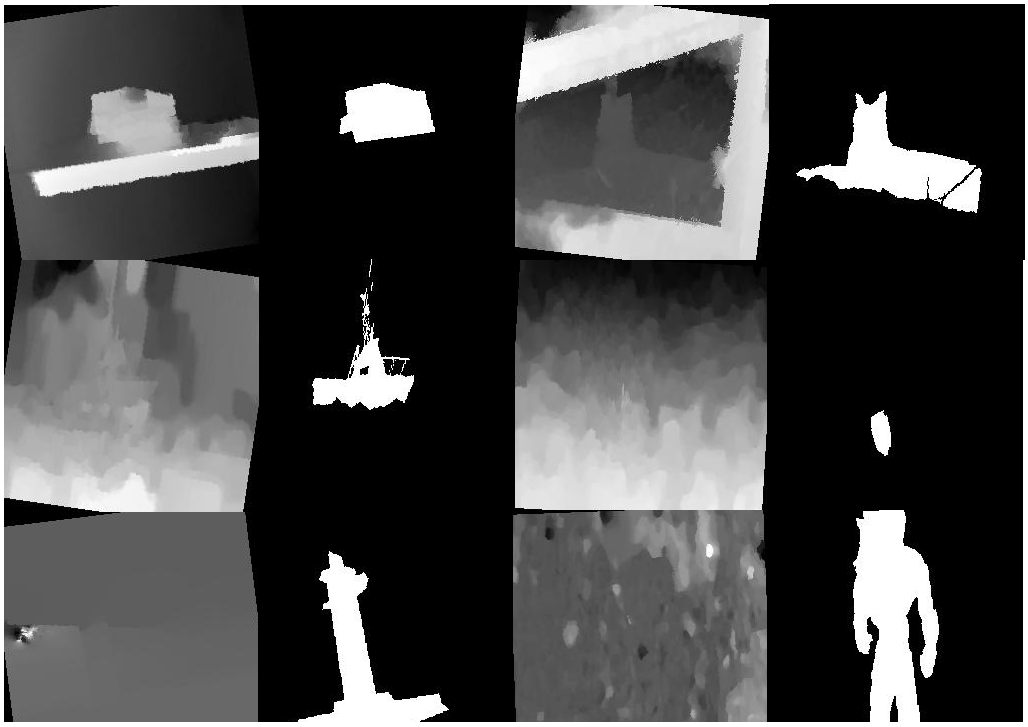


Figure 5.4: Low-quality depths in training.

salient object, other objects in depth image dominate salient features; 2) low contrast between the salient object and background in depth; 3) depth distortion caused by the camera. Intuitively, training loss is supposed to reduce drastically if the training data is distorted. Therefore we propose that these depth maps can be treated as noises when combined with RGB maps and further set an accuracy threshold during knowledge distillation to control the impact of noises:

$$\theta_{t,s} = \begin{cases} tanh({\alpha_t}^p \cdot {\beta_s}^{1-p}) & \alpha_t > threshold \\ \epsilon & Others \end{cases} \tag{5.6}$$

where ϵ indicates a small weight. We tested three options, including 0.1, 0.01 and 0.001, and 0.01 shows the best results. α_t provides knowledge prior from the teacher network and indicates whether the depth distortion happens. Here *threshold* is set to 0.5 due to binary classification. Under this circumstance, the student model is able to know the useless training data when receiving knowledge from the teacher model.

Compared to considering one aspect or enforcing a fixed weight to the student model, our dynamic weight considers both the correctness of the teacher's knowledge and the error of the student network, which allows the student network to receive the knowledge according to the degree of difficulty of samples. $\theta_{t,s}$ varies little in the start of training phase. As for the late stage of training, the student network is able to detect most simple scenarios except for some hard samples. Therefore, $\theta_{t,s}$ automatically assigns relatively bigger weights for hard samples which can be detected accurately in the teacher network but student network. The noise elimination method takes full merit of the knowledge prior from the teacher network and effectively reduces the negative impact of depth maps in low quality. Extensive experiments demonstrate in section 4 that this dynamic knowledge distillation could boost the detection performance without increasing extra parameters and model size. The process of the proposed methods is illustrated as algorithm 2

.

Algorithm 2 Dynamic Knowledge Distillation.

Input: P_t is the prediction of teacher network,
P_s is the prediction of student network,
G is the corresponding ground truth.

Stage 1: Training the teacher network;
$loss = L_{CE}(P_t, G)$
Stage 2: Training the student network;
$\alpha_t = IOU(P_t, G)$;
$\beta_s = 1 - IOU(P_s, G)$;
if $\alpha_t > Threshold$ **then**
    $\theta_{t,s} = tanh({\alpha_t}^p \cdot {\beta_s}^{1-p})$
else
    $\theta_{t,s} = 0.01$
end if
$loss = \theta_{t,s} L_{KL}(P_s, P_t) + (1 - \theta_{t,s}) L_{CE}(P_s, G)$;

5.4 Experiments

5.4.1 Datasets and Evaluation Metrics

Datasets. Extensive experiments are conducted on five widely used RGB-D datasets, namely, NLPR [30], NJUD [29], SIP [33], DES [32] and LFSD [31]. These datasets contain large-scale images with different resolutions and diverse scenarios. We adopt the same training dataset with [3], which contains 1,500 samples from NJUD and 700 samples from NLPR. The rest images in these two datasets together with the other three datasets are used for testing.

Evaluation Metrics. We adopt six metrics to comprehensively evaluate SOD tasks. These metrics include the Precision-Recall (PR) Curve, the F-measure curves, the F-measure score (2.37, the Mean Absolute Error (2.39), the S-measure (2.40) and the E-measure (2.49).

5.4.2 Implementation Details

Our model is implemented using Pytorch Toolbox and trained on a GTX TITAN X GPU for 40 epochs with mini-batch size 4. We use a VGG16- and VGG19-based FPN as our final student architecture. Both RGB and depth images are resized to 256×256. To avoid overfitting, simple flipping and rotating are adopted to augment the training dataset. The initial learning rate is set to 1e-3 and we adopt a 0.0005 weight decay for the stochastic gradient descent (SGD) with a momentum of 0.9.

5.4.3 Comparisons with the state of arts

We conduct our experiments on 13 different prevalent methods in recent years, including DF [97], CTMF [101], AFNet [98], MMCI [99], DMRA [34], CPFP [3],

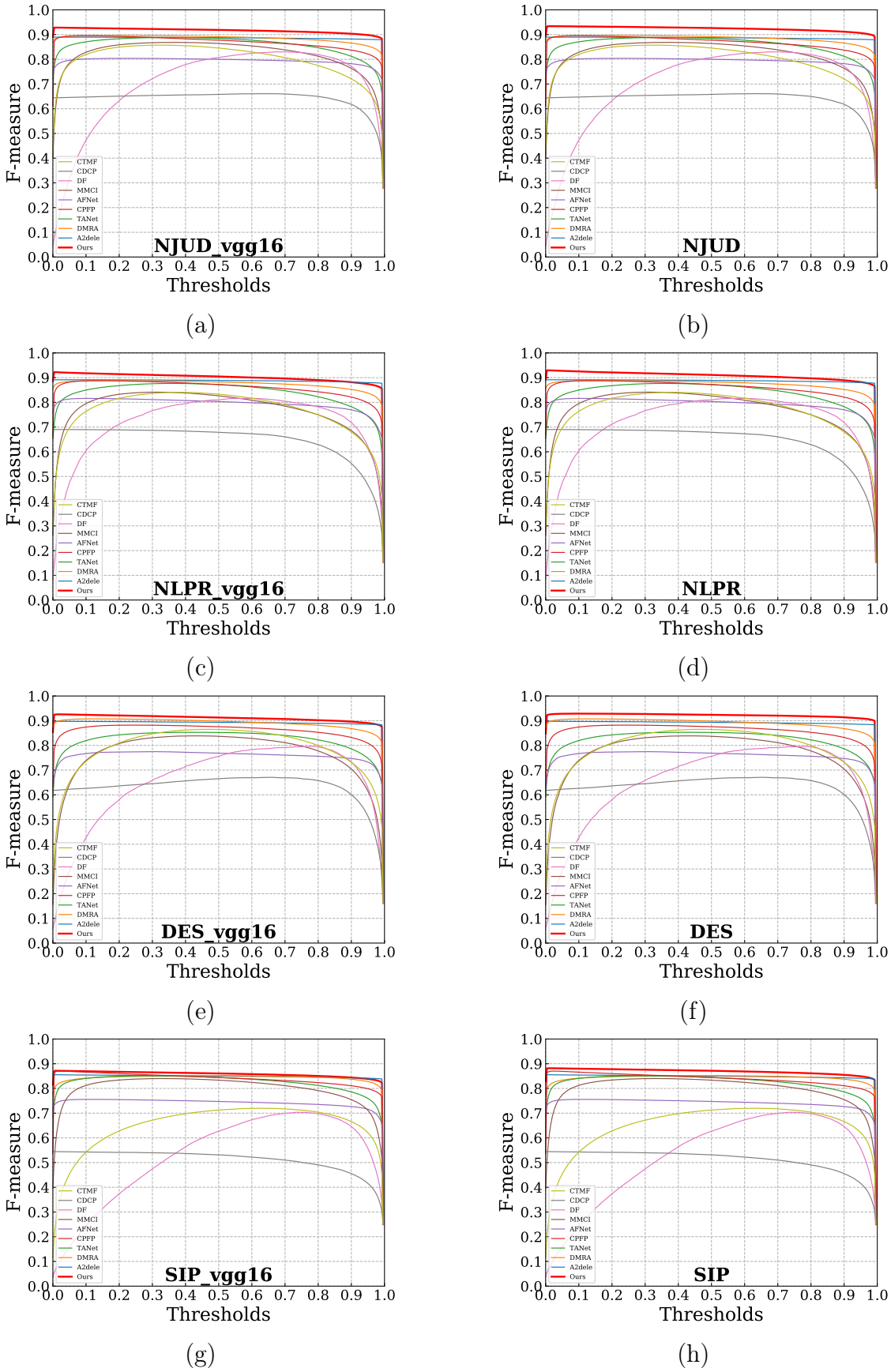


Figure 5.5: Fm curves across four benchmarks on two different backbones.

Table 5.1: Quantitative comparisons through the maximum of F-score F_β, S-score S_α, E-score E_θ, and error-score M, over five widely evaluated datasets. DKD and $DKD*$ indicate simple VGG16-based and VGG19-based FPN respectively. *ite* refers to the training iterations.

	Metric	DF	CTMF	AFNet	MMCI	TANet	DMRA	CPFP	D3Net	A2dele	DANet	FANet	CMINet	PMSFNet	DKD	DKD*
NJUD	$F_\beta\uparrow$	0.789	0.857	0.804	0.868	0.888	0.896	0.890	0.903	0.905	0.890	0.892	0.925	0.919	0.928	0.934
	$S_\alpha\uparrow$	0.735	0.849	0.772	0.859	0.878	0.885	0.878	0.895	0.867	0.897	0.899	0.939	0.910	0.916	0.920
	$E_\theta\uparrow$	0.818	0.866	0.847	0.882	0.909	0.920	0.900	0.901	0.914	0.926	0.914	0.956	0.926	0.949	0.952
	$M\downarrow$	0.151	0.085	0.100	0.079	0.061	0.051	0.053	0.051	0.052	0.046	0.044	0.032	0.045	0.032	0.030
NLPR	$F_\beta\uparrow$	0.752	0.841	0.816	0.841	0.876	0.888	0.884	0.904	0.891	0.908	0.885	0.909	0.924	0.922	0.930
	$S_\alpha\uparrow$	0.769	0.860	0.799	0.856	0.886	0.898	0.884	0.906	0.889	0.908	0.913	0.941	0.923	0.921	0.924
	$E_\theta\uparrow$	0.840	0.869	0.884	0.872	0.926	0.942	0.920	0.934	0.937	0.945	0.951	0.964	0.942	0.958	0.960
	$M\downarrow$	0.110	0.056	0.058	0.059	0.041	0.031	0.038	0.034	0.031	0.031	0.026	0.019	0.030	0.022	0.021
DES	$F_\beta\uparrow$	0.625	0.865	0.775	0.839	0.853	0.906	0.882	0.917	0.897	0.916	0.874	0.926	0.933	0.926	0.928
	$S_\alpha\uparrow$	0.685	0.863	0.770	0.848	0.858	0.899	0.872	0.904	0.883	0.905	0.894	0.953	0.926	0.918	0.918
	$E_\theta\uparrow$	0.806	0.911	0.874	0.904	0.919	0.944	0.927	0.956	0.918	0.961	0.925	0.970	0.964	0.965	0.966
	$M\downarrow$	0.131	0.055	0.068	0.065	0.046	0.030	0.038	0.030	0.030	0.028	0.026	0.015	0.027	0.022	0.023
LFSD	$F_\beta\uparrow$	0.854	0.815	0.780	0.813	0.827	0.872	0.850	0.849	0.858	-	0.855	0.862	0.894	0.865	0.862
	$S_\alpha\uparrow$	0.786	0.796	0.738	0.787	0.801	0.847	0.828	0.832	0.833	-	0.850	0.877	0.874	0.834	0.839
	$E_\theta\uparrow$	0.841	0.851	0.810	0.840	0.851	0.899	0.867	0.860	0.875	-	0.882	0.911	0.907	0.883	0.883
	$M\downarrow$	0.142	0.120	0.133	0.132	0.111	0.076	0.088	0.099	0.077	-	0.076	0.064	0.072	0.080	0.078
SIP	$F_\beta\uparrow$	0.704	0.720	0.756	0.840	0.851	0.847	0.870	0.882	0.855	0.901	-	0.887	0.913	0.872	0.882
	$S_\alpha\uparrow$	0.653	0.716	0.720	0.833	0.835	0.800	0.850	0.864	0.828	0.878	-	0.894	0.896	0.855	0.865
	$E_\theta\uparrow$	0.794	0.824	0.815	0.886	0.894	0.858	0.899	0.903	0.890	0.914	-	0.933	0.923	0.908	0.914
	$M\downarrow$	0.185	0.139	0.118	0.086	0.075	0.088	0.064	0.063	0.070	0.054	-	0.044	0.051	0.060	0.056
	Backbone	VGG16	VGG16	VGG16	VGG16	VGG16	VGG19	VGG16	VGG16	VGG16	VGG16/19	VGG16	ResNet50	VGG19	VGG16	VGG19
	Epoch	-	-	-	30,000(ite)	-	50	10,000(ite)	30	50	40	40	100	40	40	40

Method	MMCI	TANet	PCANet	D3Net	PMSFNet	CPFP	DMRA	DANet	CMINet	A2dele	Ours
Model Size(MB)	951.9	929.7	533.6	519	364.2	278	238.8	106.7	84	57.3	57.9/78.2
FPS	19	-	15	-	11	7	10	32	10	120	136

Table 5.2: The model size and inference speed of different methods.

TANet [100], D3Net [33], A2delde [91], DANet [10], FANet [107], PMSFNet [108] (Chapter 4) and CMINet [109]. For fair comparisons, we directly use the released evaluation results or generate the results by the public saliency maps under the same evaluation framework.

Quantitative Evaluation. Table 5.1 shows the quantitative results over five datasets. It can be observed that our method achieves the best scores in most metrics, especially on the NJUD dataset which contains 500 testing image pairs, for which our method performs better as far as all metrics are concerned. As for LFSD and SIP, although higher results come from other methods, we still obtain competing results in smaller VGG16 and VGG19-based networks. Figure 5.5 shows the comparison results using one-dimensional curves. Our method which is represented by the red line demonstrates better overall performance in both lightweight student models. Besides, it is apparent in Table 5.2 that our final VGG16-based network only has 57.9MB with a faster speed, which drastically improves the inference speed and reduces the number of parameters. The above results indicate that without

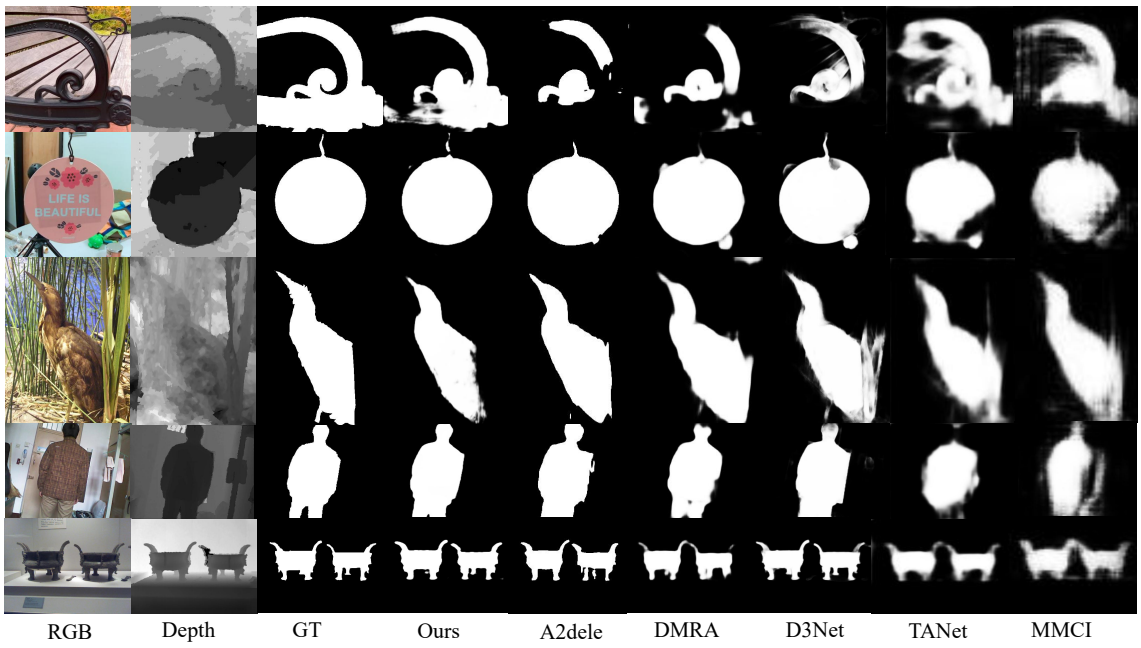


Figure 5.6: Visual comparisons with existing methods

designing complicated models, accurate detection results can be obtained by only using an FPN with the help of the proposed methods.

Qualitative Evaluation. Figure 5.6 exhibits the visual comparisons with prevalent methods in recent years. Images contain diverse objects and scenarios, which are picked from different testing datasets. It can be observed that the saliency maps generated by our method are closer to the ground truth. More specifically, row 1 shows the case where the depth image has low contrast, especially on the bottom part, and rows 2 and 3 show complex backgrounds in the RGB images. Under these circumstances, our method generates better saliency maps with less distortion and irrelevant objects compared to other methods.

5.4.4 Ablation Studies

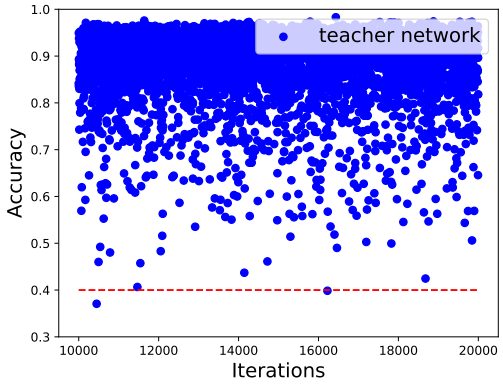
Dynamic Knowledge Distillation. As shown in Table 5.3, our baseline is an FPN with a VGG19 backbone trained on a cross-entropy loss, which can achieve the basic detection task. *RGB* indicates that only using RGB maps in training and *RGBD*

Table 5.3: Ablation analysis on 4 datasets. *RGB* and *RGBD* indicate that the student network is trained without/with depth maps respectively by cross-entropy loss. s indicates the weight of knowledge distillation. Here we use the mean value of F-score F_m and wF_m to show the overall accuracy.

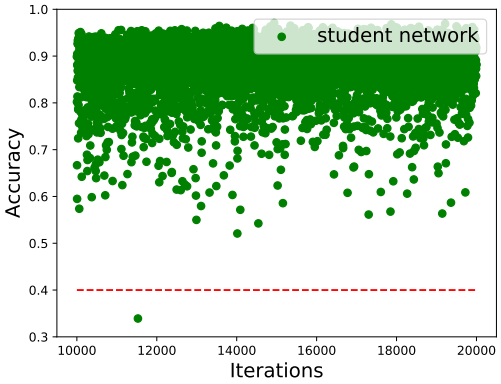
	Metric	RGB	RGBD	s=0.3	s=0.5	s=0.7	s=Dynamic	+threshold
SIP	$F_m \uparrow$	0.704	0.773	0.832	0.845	0.843	0.849	**0.853**
	$wF_m \uparrow$	0.654	0.724	0.796	0.805	0.809	0.811	**0.817**
	$M \downarrow$	0.108	0.086	0.063	0.061	0.059	0.058	**0.056**
NJUD	$F_m \uparrow$	0.776	0.830	0.902	0.902	0.898	0.904	**0.914**
	$wF_m \uparrow$	0.739	0.799	0.895	0.889	0.880	0.893	**0.901**
	$M \downarrow$	0.080	0.060	0.030	0.034	0.037	0.032	**0.030**
NLPR	$F_m \uparrow$	0.780	0.816	0.873	0.876	0.870	0.876	**0.890**
	$wF_m \uparrow$	0.746	0.781	0.877	0.875	0.865	0.876	**0.887**
	$M \downarrow$	0.046	0.041	0.022	0.024	0.026	0.024	**0.021**
LFSD	$F_m \uparrow$	0.713	0.784	0.825	0.832	0.830	0.834	**0.835**
	$wF_m \uparrow$	0.656	0.741	0.780	0.793	0.790	0.795	**0.796**
	$M \downarrow$	0.142	0.102	0.086	0.080	0.080	0.078	**0.078**

concatenates depth maps as input. It is worth noting that directly using an early fusion strategy also shows potential in RGB-D saliency detection. Then we employ KD on the baseline to compare the results in different weights on four datasets. It is observed that KD can improve the performance and our dynamic KD achieves better results across four testing datasets. Furthermore, Figure 5.7 shows the detection performance of the student network in different KD weights. It is observed that in the last 10,000 iterations of the training stage, the proposed method has better overall accuracy, where the lowest accuracy is still above 0.4, which is even better than the teacher network. Specific examples in Figure 5.8 illustrate that compared with the dynamic KD, conventional fixed weights suffer from more false positives and negatives. Consequently, it is demonstrated that our dynamic KD adaptively controls the knowledge distillation in an appropriate way, leading to the improvement of overall detection performance.

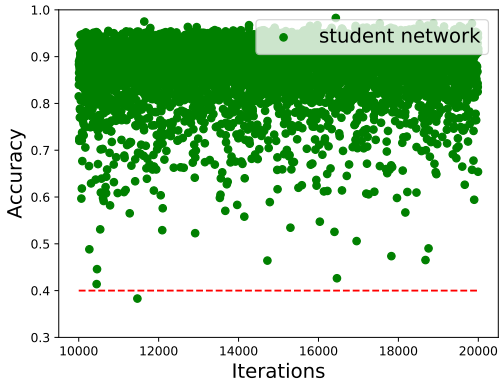
Noise Elimination With Dynamic Knowledge Distillation. We investigate the low accuracy issue in the teacher network by visualizing extremely hard samples which have been shown in Figure 5.4. It is observed in Table 5.3 that with the help of noise elimination, all evaluation metrics over four testing datasets approach better results.



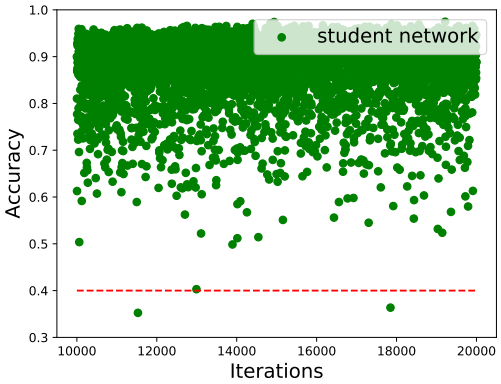
(a) teacher



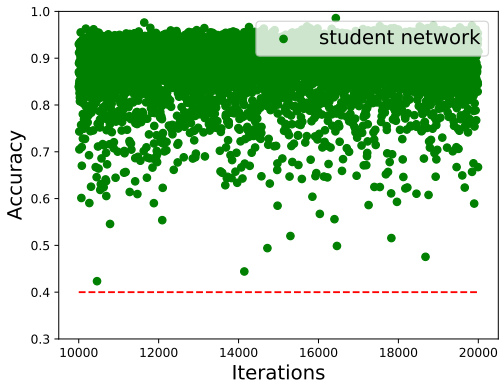
(b) 0.3



(c) 0.5



(d) 0.7



(e) dynamic

Figure 5.7: Teacher and student scatters with different distillation weights. We exhibit the last 10,000 training iterations and dynamic KD shows better overall performance. It is worth noting that using 0.3 suffers from the overfitting issue (Table 5.3).

Red arrows and rectangles in Figure5.9 label the details which are refined by the proposed methods, especially on the part of the object in the low contrast background, further illustrating that the proposed noise elimination effectively mitigates the noise

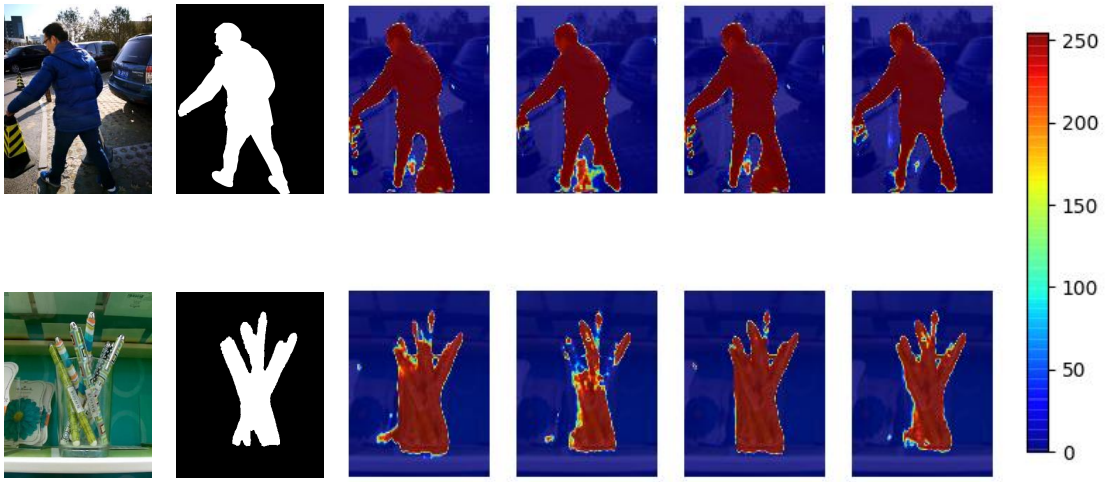


Figure 5.8: Heatmap visual comparisons of different distillation weights: heatmaps from left to right indicate weights 0.3, 0.5, 0.7 and dynamic respectively.

during distillation and make the student model be able to learn more semantic details in the useful training data.

Table 5.4: Quantitative comparisons of applying DKD on DANet.

	Metric	DANet	DANet+DKD
SIP	$F_m \uparrow$	0.864	0.845
	$wF_m \uparrow$	0.829	0.808
	$M \downarrow$	0.054	0.059
DES	$F_m \uparrow$	0.891	0.891
	$wF_m \uparrow$	0.848	0.870
	$M \downarrow$	0.028	0.024
	Model Size(MB)	106.7	78.2

.

Further Analysis. In order to show the generalization of the proposed DKD, we replace the teacher network with DANet. Experimental results in Table 5.4 indicate that our DKD can compress the model size of the existing method and achieve close accuracy to the teacher model. We further conduct experiments on VGG16-based FPN with different KD hyper-parameters as shown in Table 5.5. Specifically, we

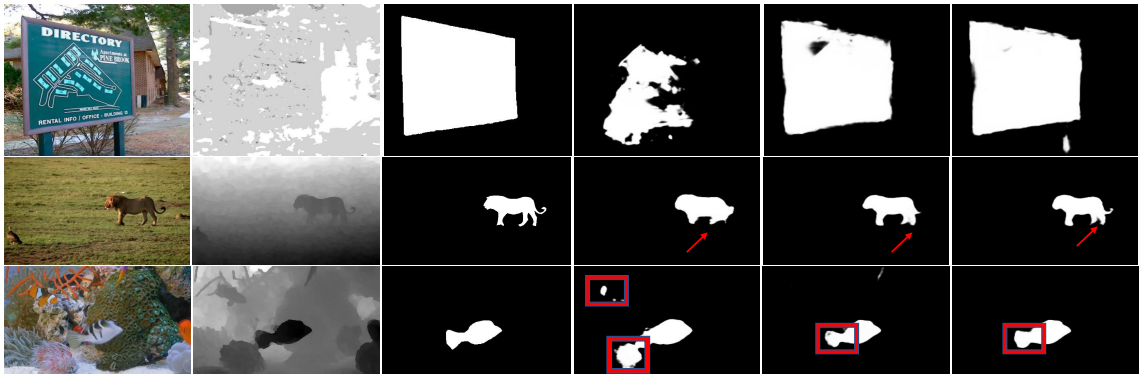


Figure 5.9: Ablation studies of the proposed methods. *Baseline* represents that the student network is only trained on cross-entropy loss. DKD represents the proposed dynamic knowledge distillation and NE means noise elimination.

set the temperature to 10 and only use the RGB images in the distillation training phase. Experimental results demonstrate that the proposed DKD can be utilized on different networks with different training settings, proving the effectiveness and generalization of DKD and leading to the potential of achieving RGB-D SOD tasks through RGB data within a lightweight structure.

Table 5.5: Distillation on VGG16 with RGB maps.

	Metric	s=0.3	s=0.5	s=0.7	s=Dynamic
NLPR	$F_m \uparrow$	0.882	0.876	0.876	**0.884**
	$wF_m \uparrow$	0.869	0.865	0.866	**0.879**
	$M \downarrow$	0.024	0.025	0.027	**0.024**
LFSD	$F_m \uparrow$	0.790	0.786	0.788	**0.798**
	$wF_m \uparrow$	0.745	0.735	0.751	**0.753**
	$M \downarrow$	0.104	0.106	0.101	**0.097**

5.5 Summary

In this chapter, we propose a dynamic knowledge distillation strategy and a noise elimination method for RGB-D based SOD. The proposed dynamic strategy considers the performance of both teacher and student models to generate an adaptive weight for knowledge distillation. Besides, we investigate the noise issue caused by depth maps and alleviate this problem by setting a threshold in KD. The proposed methods

provide a new perspective that avoids designing extra networks for RGB-D SOD. We conduct comprehensive experiments on five challenging benchmark datasets to demonstrate that our method achieves competitive performance by only using a simple FPN model, which significantly compresses the model size and increases the inference speed. We further apply this dynamic strategy on different distillation temperatures with diverse models to prove the effectiveness and generalization of our method.

Chapter 6

Adaptive Intra-Group Aggregation for Co-Saliency Detection

6.1 Introduction

As a new attractive branch of the SOD tasks, co-salient object detection (CoSOD) aims at detecting common foreground and co-occurring objects in a group of images. Due to the potential use of detecting objects in multiple relevant images, CoSOD can be treated as a pre-processing step and has been widely employed in diverse computer vision tasks, such as image retrieval [110], image surveillance [111], co-segmentation [112] and weakly-supervised segmentation [113]. One of the key challenging issues of this task is modeling the relationship between individual feature representation in a single image and group feature representation. Wei [114] investigates the interaction relationship between single image and group images by processing them separately in different network blocks to obtain the unique characteristics of every single image and group consistency. Zhang [115] extracts the intra- and inter-image correspondence of an image group by designing an adaptive graph convolutional network. Both these methods model the interaction relationship in suboptimal ways due to the requirement of extra parameters and networks. This

issue motivates us to propose a simple but effective algorithm that can formulate the discrepancy between the group and individual feature representations and meanwhile adaptively fuse these features. In addition, due to the size variation among different salient objects, we propose a scale-aware loss to further refine the detection performance.

6.2 Related Works

6.2.1 Co-Salient Object Detection

Accurate detection performance has been achieved on deep learning methods, which jointly learn co-salient object representations [35]. Zha [116] proposes a new end-to-end approach to learning single-image features with pyramid spatial attention operation. Subsequently, a hierarchical low-rank bilinear pooling function is employed to learn group semantic representation for co-category classification. Li [117] designs a co-attention recurrent unit to model the final group representation, which is treated as synergetic information and fused with each individual image.

6.2.2 Determinantal Point Processes

In this work, determinantal point processes (DPPs) are used to model the diversity within the intra-group. Recently, DPPs have been widely used in the machine learning community, such as video summarization [118], recommendation systems [119], and deep reinforcement learning [120, 121].

In general, for a discrete set of points $\mathcal{Y} = \{x_1, x_2, \cdots, x_N\}$, a point process $\mathcal{P}$ is a probability measure over all $2^{|\mathcal{Y}|}$ subsets. $\mathcal{P}$ is a DPP if a random subset $\mathbf{Y}$ is sampled with probability:

$$\mathcal{P}_L(\mathbf{Y} = Y) \propto \det(L_Y), \tag{6.1}$$

where $Y \subseteq \mathcal{Y}$, $L \in \mathbb{R}^{N \times N}$ is the positive semi-definite DPP kernel matrix, and L_Y is the sub-matrix with rows and columns indexed by the elements of the subset Y. The kernel matrix L can be represented as the Gram matrix $L = X^T X$, where each column of X is the feature vector of an item in $\mathcal{Y}$. $\det(L_Y)$ therefore represents the (squared) volume spanned by vectors $x_i \in Y$. Especially, v_1 and v_2 refer to the feature vectors, the determinant of the Gram matrix can be expressed as:

$$|L_Y| = \begin{vmatrix} (v_1, v_1) & (v_1, v_2) \\ (v_2, v_1) & (v_2, v_2) \end{vmatrix} \tag{6.2}$$

Where (,) represents the dot product between two feature vectors. From a geometrical perspective, feature vectors in a subset that are more orthogonal will have a larger determinant, increasing their probability of being sampled in a DPP: $\mathcal{P}_L(\mathbf{Y} = Y) \propto \det(L_Y)$. Using orthogonality as a measure of diversity, we adopt DPPs to model the diversity of the samples in the intra-group.

The overall architecture is shown in Figure 6.1. We simply adopt the Feature Pyramid Network (FPN) [18] structure with a VGG-16 [122] backbone as our final network. Considering the computing resources, the proposed Adaptive Intra-Group Aggregation (AIGA) modules are only applied on deep layers, namely, $Conv5'$, $Conv4'$ and $Conv3'$.

6.3 Methodology

6.3.1 Adaptive Intra-Group Aggregation

As aforementioned above, previous approaches require extra networks and parameters to formulate and aggregate group-wise and individual features. This issue motivates us to propose a simple but effective algorithm, which can formulate the discrepancy between the group and individual feature representations and meanwhile adaptively

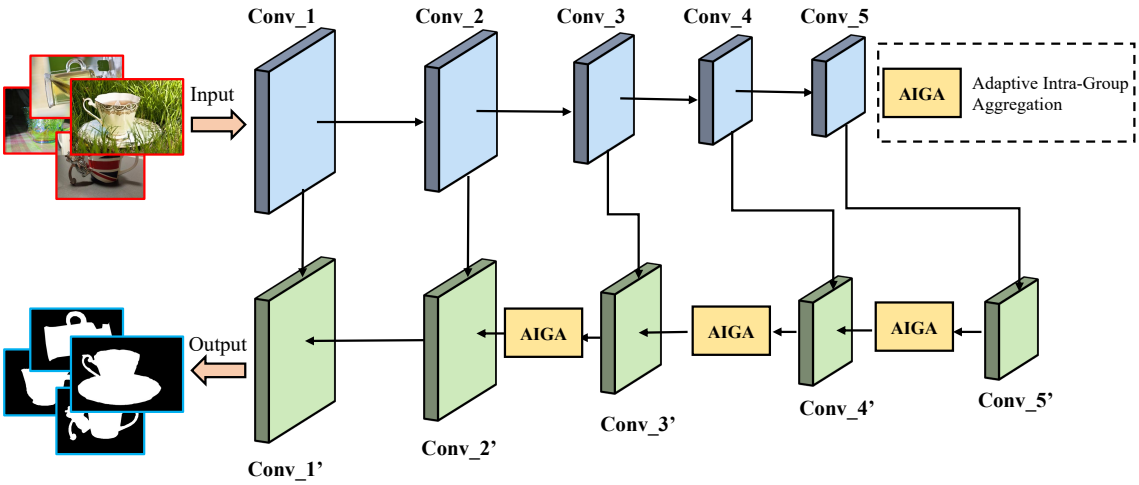


Figure 6.1: Overall structure of the proposed method.

fuse these features. We are given a group of images $G = \{I_n\}_{n=1}^{N}$ as the input for the encoder network $F(\cdot)$, such as a VGG-16 backbone. This network extract features from group images and the feature representation of each image can be obtained:

$$e_n = F(I_n) \tag{6.3}$$

Then we simply concatenate individual feature representations together and adopt a mean operation to effectively obtain the group feature representation:

$$E = \frac{1}{N}\sum_{n=1}^{N} e_n \tag{6.4}$$

where E indicates the group feature representation of N images. Then we calculate the diversity between the group G and each subset I_n by DPP:

$$\alpha_d = DPP(e_n, E) \tag{6.5}$$

Subsequently, we treat the diversity α_d of e_n and E as the discrepancy between group features and individual features and adaptively fuse group features with each image according to the discrepancy:

$$\hat{e_n} = e_n + \alpha_d \cdot E \tag{6.6}$$

where $\hat{e_n}$ indicates the final individual feature representation. With the proposed AIGA, the processed individual features not only contain unique characteristics of the single image but the common foreground information of group images.

6.3.2 Scale-aware Loss



Figure 6.2: Different sizes of objects from different categories.

Objects from diverse categories have different sizes. Examples from Figure 6.2 show the size discrepancy between an alarm clock and a hammer. It can be observed that the hammer has a bigger size while the clock only takes a small portion of the whole image. Thus, to better capture the scale variation of salient objects among different groups, we design a group-wise scale-aware loss by calculating the scale prior to each group. Concretely, the scale prior can be obtained as follows:

$$S = \frac{1}{N} \sum_{n=1}^{N} GT_n \tag{6.7}$$

where GT_n refers to the corresponding ground truth of I_n. Therefore, S indicates the scale prior to a specific group with N images. Due to the imbalance of objects' sizes, the model should treat each group unequally in the training phase. More specifically, a large punishment should be assigned on training loss when S is small and vice

versa. The adaptive punishment weight of each group is formulated as follows:

$$\beta_g = (1 - S)^p \tag{6.8}$$

Here we use $p = 0.1$ to control the variation of β_g. Finally, the training loss can be expressed as:

$$L_{iou}(GT_n, P) = \beta_g \cdot L_{iou}(GT_n, P) \tag{6.9}$$

where P is the prediction of each image in group G, L_{iou} refers to IOU loss function. To this end, the model is able to effectively capture the relative scale information of salient objects in a specific group.

6.4 Experiments

6.4.1 Datasets and Evaluation Metrics

We adopt three current challenging datasets for evaluating the effectiveness of the proposed algorithms: CoCA [11], CoSOD3k [35] and Cosal2015 [36]. In order to comprehensively evaluate the detection results, we use four prominent evaluation metrics, namely, the F-measure (2.37), Mean Absolute Error (2.39), E-measure (2.49) and S-measure (2.40).

6.4.2 Implementation Details

We adopt FPN with a VGG-16 backbone as our baseline. We use DUTS[24] dataset in the training phase and follow the same group setting in [11]. The final model is trained on a GTX TITAN X GPU for 50 epochs with the Adam optimizer. The initial learning rate is set to $1e-4$ and the batch size is 16. All images are reshaped

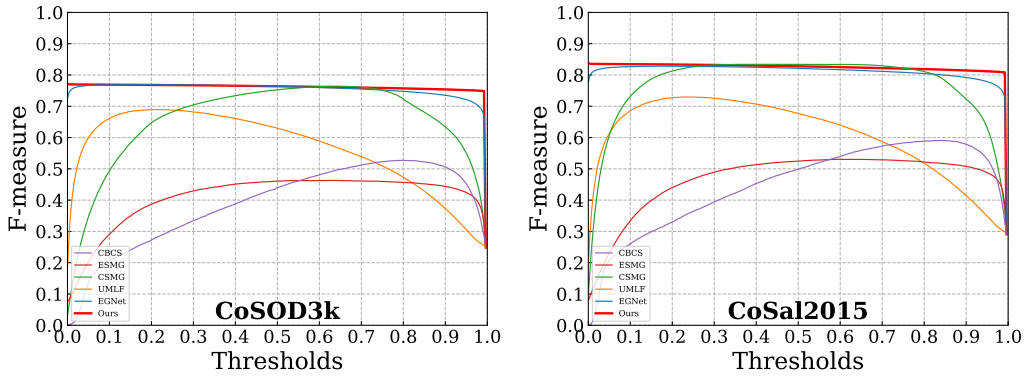


Figure 6.3: F-measure curves on two public CoSOD datasets.

to 224×224 in the training and testing stages and reshaped to the original size in the evaluation stage.

Table 6.1: Quantitative comparisons through the maximum of F-score F_β, S-score S_α, E-score E_θ, and error-score M, over three widely evaluated datasets.

	Metric	CBCS [123]	GWD [114]	RCAN [124]	ESMG [125]	BASNet [57]	UMLF [126]	CSMG [127]	EGNet [128]	SCRN [129]	Ours *
CoSOD3k	$F_\beta \uparrow$	0.466	0.649	0.688	0.418	0.720	0.639	0.709	0.702	0.716	**0.746**
	$S_\alpha \uparrow$	0.528	0.716	0.744	0.523	0.771	0.632	0.711	0.762	0.771	**0.788**
	$E_\theta \uparrow$	0.637	0.777	0.808	0.635	0.804	0.758	0.804	0.793	0.805	**0.836**
	$M \downarrow$	0.228	0.147	0.130	0.239	0.114	0.285	0.157	0.119	0.113	**0.088**
Cosal2015	$F_\beta \uparrow$	0.523	0.706	0.764	0.476	0.791	0.690	0.784	0.786	0.783	**0.812**
	$S_\alpha \uparrow$	0.544	0.744	0.779	0.522	0.822	0.662	0.774	0.818	0.817	**0.829**
	$E_\theta \uparrow$	0.656	0.802	0.842	0.640	0.849	0.769	0.842	0.843	0.850	**0.865**
	$M \downarrow$	0.233	0.148	0.126	0.247	0.096	0.271	0.130	0.099	0.098	**0.080**
CoCA	$F_\beta \uparrow$	0.313	0.408	0.422	-	0.408	-	**0.499**	0.404	0.413	0.466
	$S_\alpha \uparrow$	0.523	0.602	0.616	-	0.592	-	0.627	0.603	0.612	**0.628**
	$E_\theta \uparrow$	0.641	0.701	0.702	-	0.644	-	**0.733**	0.648	0.642	0.681
	$M \downarrow$	0.180	0.166	0.160	-	0.195	-	**0.114**	0.178	0.164	0.148

6.4.3 Performance Comparison

We compare our method with 9 challenging state-of-the-art methods, including representative traditional models and deep learning models, namely, CBCS [123], ESMG [125], UMLF [126], GWD [114], RCAN [124], CSMG [127], EGNet [128], SCRN [129] and BASNet [57]. We directly use the public saliency maps and evaluation results for fair comparisons.

Quantitative Evaluation. We show F-measure curves in Figure 6.3 to evaluate the overall performance of the CoSOD. As shown, our method which is represented by

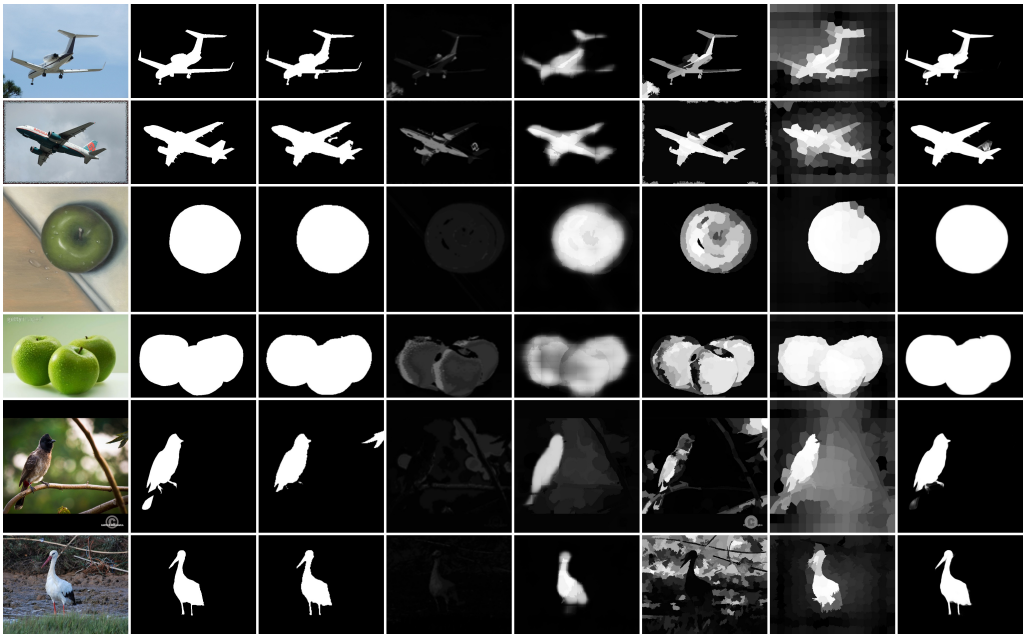


Figure 6.4: Qualitative comparisons with state-of-the-arts in different categories.

the red line outperforms other state-of-the-arts. In Table 6.1, compared with previous approaches, our algorithm achieves competitive results across all datasets, especially on CoSOD3k and Cosal2015, which show the best results on four evaluation metrics.

Compared to the traditional methods, our method shows better performance in all metrics across three datasets. Especially for ESMG, our method boosts the performance by a large margin on Cosal2015, providing gains of 33.6%, 30.7%, 22.5% and 16.7% in terms of the maximum F_β, S_α, E_θ and M respectively. In addition, compared to deep learning methods, we only exploit a simple Feature Pyramid Network (FPN) with a VGG-16 backbone without designing delicate networks. For instance, EGNet requires 240.3 GFLOPs and our model only requires 62.38 GFLOPs, which further demonstrates that our proposed method can achieve better performance with less computational complexity.

Qualitative Evaluation. Figure 6.4 shows the visual results of the proposed method and other representative methods. It can be observed that our saliency maps are more similar to the ground truth and show better performance on completeness and edges in different groups.

Table 6.2: Ablation studies on the proposed methods.

	Metric	Base	AIGA5	AIGA54	AIGA543	+Scale
CoSOD3k	$F_\beta \uparrow$	0.731	0.742	0.743	0.742	0.746
	$S_\alpha \uparrow$	0.774	0.780	0.781	0.780	0.788
	$E_\theta \uparrow$	0.825	0.832	0.833	0.834	0.836
	$M \downarrow$	0.096	0.088	0.089	0.089	0.088
Cosal2015	$F_\beta \uparrow$	0.798	0.807	0.812	0.815	0.812
	$S_\alpha \uparrow$	0.817	0.820	0.823	0.825	0.829
	$E_\theta \uparrow$	0.856	0.861	0.864	0.867	0.865
	$M \downarrow$	0.088	0.086	0.084	0.082	0.080

Ablation Analysis. We conduct experiments to show the effectiveness of each proposed algorithm. As shown in Table 6.2, *Base* refers to the FPN baseline. We apply the AIGA on different levels from deep to shallow layers. For instance, $AIGA5$ indicates we only apply the proposed AIGA on the deepest level $Conv5'$. It can be clearly observed that the proposed AIGA can effectively improve the detection results and three levels ($AIGA543$) show better results. *Scale* represents that the scale-aware loss is applied to the model, leading to further improvement of the detection results. Apparently, it is demonstrated that both proposed methods can effectively improve the accuracy without incorporating additional parameters.

Further Analysis. In Table 6.3, CBCS, ESMG and UMLF are traditional methods. BASNet, EGNet and Ours are deep-learning-based methods. From Table 6.3, our method requires less run-time and computational complexity than other baselines.

	CBCS	ESMG	UMLF	BASNet	EGNet	Ours
GFLOPs	-	-	-	195.02	240.30	62.38
Time (s)	0.3	1.2	87	1.24	0.19	0.17

Table 6.3: Comparison of GFLOPs and run-time (second) between our method and others. All models are evaluated on one Nvidia Titan XP, and the size of the input is 224×224×3.

We have shown the results without AIGA when the scale awareness is adopted in Table 6.4. As mentioned above, this scale-aware loss aims to capture the scale variation of salient objects among different groups in the training phase. In other words, this method is able to balance the training loss and relatively assign more weights to

Table 6.4: Ablation studies on the scale-aware loss.

	Metric	Base	+Scale
CoCA	$F_\beta \uparrow$	0.443	0.446
	$S_\alpha \uparrow$	0.611	0.611
	$E_\theta \uparrow$	0.672	0.675
	$M \downarrow$	0.157	0.156
CoSOD3k	$F_\beta \uparrow$	0.731	0.732
	$S_\alpha \uparrow$	0.774	0.774
	$E_\theta \uparrow$	0.825	0.826
	$M \downarrow$	0.096	0.096
Cosal2015	$F_\beta \uparrow$	0.798	0.799
	$S_\alpha \uparrow$	0.817	0.816
	$E_\theta \uparrow$	0.856	0.857
	$M \downarrow$	0.088	0.088

specific groups which have smaller salient objects. Therefore, it is a data-based method and its benefit depends on the training dataset and the hyper-parameter p. Figure 6.5 shows the distribution of scale priors S and the adaptive loss weights β_g. We analyze the minor improvement that the hyper-parameter p squeezes the scale variation among different groups and makes the weights β_g have subtle differences in the training phase. Table 6.4 still shows the effectiveness of the proposed scale-aware loss. We further conduct experiments on different diversity computational methods, namely dot product and DPP. DPP in Table 6.5 shows better detection accuracy across all evaluation metrics over three benchmarks, demonstrating the superiority of the DPP method.

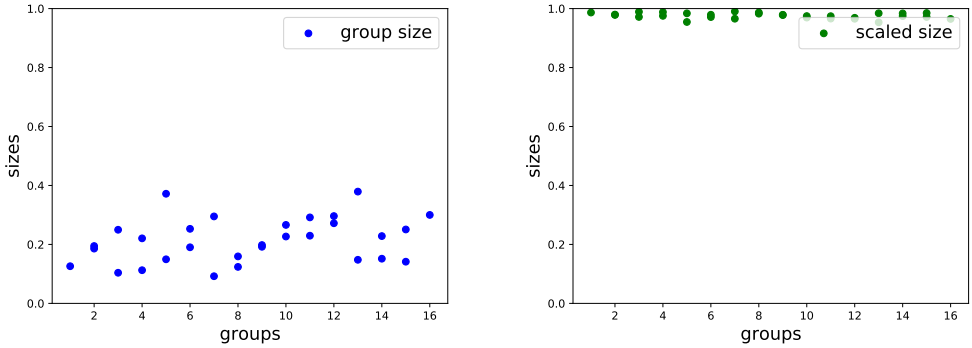


Figure 6.5: The left scatter shows the S values in the group-wise scale priors. The right shows the adaptive β_g values in the scale-aware loss. We analyze that the negligible difference in weights β_g leads to minor improvement.

Table 6.5: Comparisons of different diversity computation methods.

	Metric	dot	Ours
CoCA	$F_\beta \uparrow$	0.457	0.461
	$S_\alpha \uparrow$	0.618	0.622
	$E_\theta \uparrow$	0.687	0.695
	$M \downarrow$	0.146	0.140
CoSOD3k	$F_\beta \uparrow$	0.740	0.744
	$S_\alpha \uparrow$	0.780	0.781
	$E_\theta \uparrow$	0.832	0.834
	$M \downarrow$	0.090	0.088
Cosal2015	$F_\beta \uparrow$	0.805	0.809
	$S_\alpha \uparrow$	0.820	0.820
	$E_\theta \uparrow$	0.861	0.862
	$M \downarrow$	0.085	0.085

6.5 Summary

In this chapter, we first propose a novel feature aggregation algorithm for co-saliency detection. Our adaptive intra-group aggregation utilizes a determinantal point process to formulate the discrepancy between group features and individual features and simply combine these features in an adaptive way. We also propose a scale-aware loss to capture the scale variation and provide scale prior to each group to the model. Extensive experiments demonstrate that both methods can effectively improve detection performance.

Chapter 7

Conclusion

7.1 Summary of Thesis Achievements

This thesis has been concerned with four challenging problems of salient object detection, including RGB based, RGB-D based, and co-saliency detection. The primary focus is the development of a deep-learning detector that can detect target objects with clear boundaries in complex scenes. The research in this thesis shows the potential that designing novel networks and making full use of features leads to satisfactory saliency maps.

First, in order to mitigate the false positive and negative issues in RGB based SOD, the Pyramid Self-Attention Module (PSAM) has been proposed. It exploits the multi-scale structure and introduces the self-attention mechanism in parallel. Thus, this PSAM is able to enhance high-level semantic features on multiple scales and help the model generate better predictions with fewer false negatives. In addition to spatial enhancement, channel-wise attention modules are further used to reduce noise on different scales when features pass through the top-down pathway of the FPN baseline. The proposed structure achieves state-of-the-art on five benchmarks.

Second, to make full merit of the two-stream structure and alleviate the depth noise

in RGB-D based SOD, the Mask-Guided Feature Aggregation (MGFA) has been proposed. It introduces a mask-guided mechanism to filter irrelevant noise caused by low-quality depths in cross-domain feature fusion. With the aid of MGFA, the two-stream baseline can effectively fuse RGB and depth features at multiple scales. Similar to the proposed MGFA, the Mask-Guided Refinement Module (MGRM) shares the same mask-guided mechanism to filter the multi-scale noise from deep to shallow layers. Thus, the final framework can progressively generate accurate saliency maps and achieves state-of-the-art.

Third, considering large models in RGB-D SOD, a new dynamic knowledge distillation method has been proposed to reduce the model size and maintain detection accuracy. This method takes both the knowledge confidence of the teacher model and the knowledge hardness of the student model into consideration to generate a dynamic weight that controls the knowledge transfer process. In addition, a simple noise elimination method has been proposed to alleviate the impact of depth noise in the training phase. Extensive experiments have demonstrated that with the equipment of the proposed methods, competitive performance can be achieved in smaller and faster networks.

Furthermore, an adaptive features fusion method based on determinantal point processes has been developed. It exploits the diversity computation to measure the discrepancy between individual and group features and fuse them in an adaptive way. This method effectively enhances the single image features and can be exploited on multiple scales according to computational resources. Additionally, to balance the group sizes in different categories, a simple scale-aware loss has been proposed to help the model relatively allocate larger weights on small objects. It has been demonstrated that both methods can improve detection performance without increasing networks.

In general, the success of the proposed frameworks demonstrates that the research in this thesis provides effective methods for diverse saliency detection tasks. In particular, the well-designed training strategy and algorithm can reduce the model size and

meanwhile approach good detection performance. To this end, the methodologies in this thesis provide the potential for practical use.

7.2 Potential Applications

Animals attempt to adapt their appearance to fit the environment to avoid recognition during the evolution process. This behavior is called background-matching camouflage. Figure 7.1 shows examples of camouflaged animals in the natural environment, especially the dog and the stick insect, which are seamlessly embedded in the background.

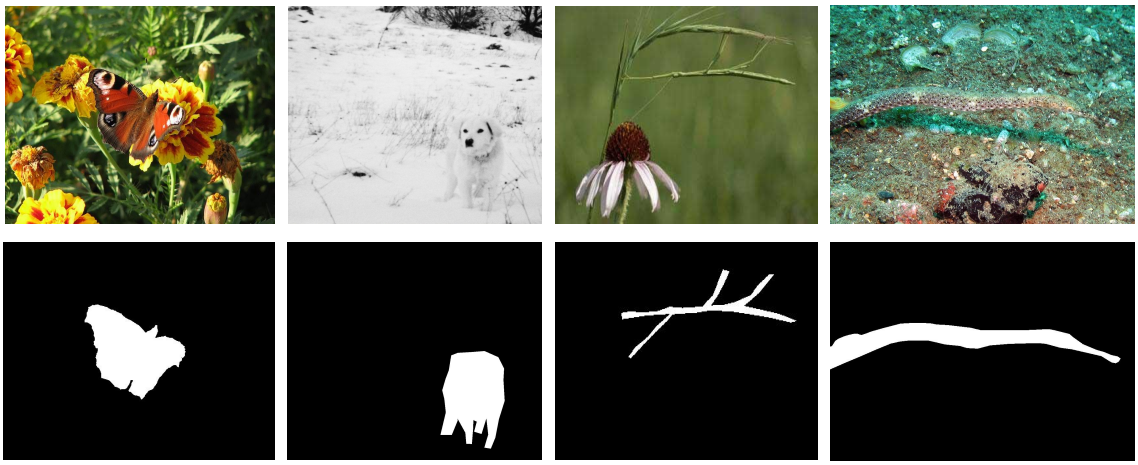


Figure 7.1: Concealed objects and the corresponding ground truth

Recently, concealed object detection as a new subset of SOD has attracted more attention. This task focuses on special conditions in which the target objects have high intrinsic similarities to the background and aims to identify them. Thus, this task can be exploited for concealed animal detection for finding and protecting rare species.

Besides, it is well known that early diagnosis through medical imaging plays a key role in the treatment of diseases. However, the early disease area usually shares similar attributes such as color and texture with its surrounding area as shown in Figure 7.2. This phenomenon leads to difficulty identify the lesion area in the early

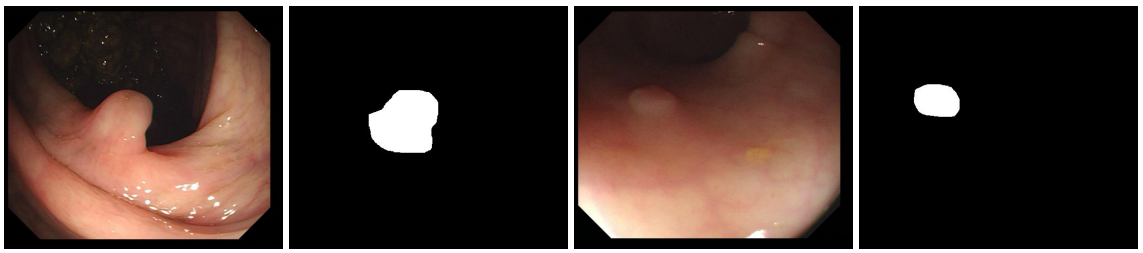


Figure 7.2: Polyp images and ground truth in medical segmentation.

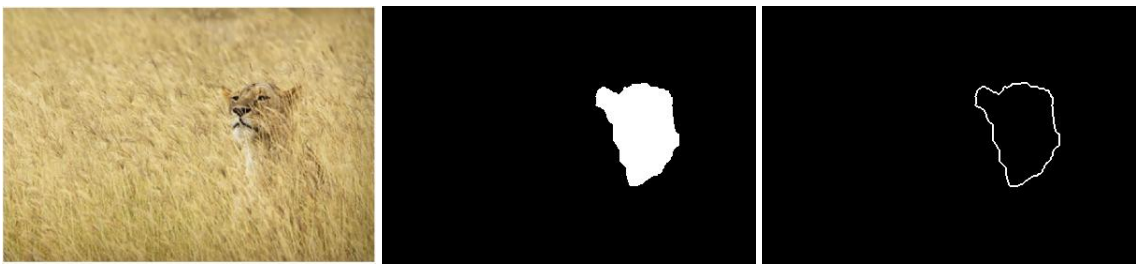
stage from a medical image. Fortunately, salient and concealed object detection provide potential utilization to tackle medical segmentation problems.

7.3 Future Work

The saliency detection algorithms in this thesis focus on improving detection accuracy and efficiency. There are still multiple directions that can be further explored for future research.

Diverse tasks. Chapter 5 proposes a dynamic knowledge distillation for saliency detection. Extensive experiments have demonstrated that this dynamic distillation weight outperforms the fixed distillation weights. However, knowledge distillation is a general method for various computer vision tasks. Therefore, this dynamic strategy should be explored in different related tasks, such as object segmentation and detection, in the future to further demonstrate the generalization ability.

Combination with edges. All the methods proposed in this thesis are based on a single-task learning framework. Inspired by the human learning process, multi-task learning has been proposed to improve performance by simultaneously learning multiple related tasks. Therefore, models are able to benefit from extra knowledge and restraint. As mentioned in this thesis, saliency detection suffers from blurred boundaries issue caused by low contrast between foreground and background. Thus, edge information plays a vital role in saliency detection. Followed by the multi-task learning strategy, edge detection can be treated as a related task and combined with



RGB GT Edge GT

Figure 7.3: Examples of an input image and the corresponding labels.

saliency detection.

Figure 7.3 shows the extra edge ground truth which can be obtained by the original ground truth. In general, existing multi-task frameworks build an extra branch for edge detection and jointly train two branches with corresponding labels. This joint training method indeed improves the accuracy of saliency detection but also increases the parameters. Utilizing the edge information in an effective way needs to be explored.

Video saliency detection. Video saliency detection aims at successively segmenting the motion-related salient object from given video sequences by taking both the spatial and temporal information into consideration. Similar to co-saliency detection, this task processes multiple images simultaneously and extracts spatial information from individual frames and temporal information from the inter-frame constraints. The differences between co-saliency detection and video saliency detection lie in two aspects: the temporal property in inter-frame correspondence and the motion cues information. Therefore, this task can be derived from co-saliency detection by integrating the motion cues. Based on Chapter 6, we can develop the model on video saliency detection and make improvements.

Different supervision strategies. Current deep SOD methods typically use a fully supervised strategy to train saliency predictors with pixel-level labels, which are annotated through a tedious and time-consuming procedure. Inspired by the human learning strategy, which usually learns with little or even no supervision, weakly-

supervised and unsupervised training methods have been proposed and developed. This limited supervision method has been introduced in SOD by leveraging category-level labels or pseudo pixel-wise annotations for improving detection performance. However, there is still a notable gap with the fully-supervised methods.

Bibliography

[1] J.-J. Liu, Q. Hou, M.-M. Cheng, J. Feng, and J. Jiang, "A simple pooling-based design for real-time salient object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3917–3926, 2019.

[2] G. Ren, T. Dai, P. Barmpoutis, and T. Stathaki, "Salient object detection combining a self-attention module and a feature pyramid network," *Electronics*, vol. 9, no. 10, p. 1702, 2020.

[3] J.-X. Zhao, Y. Cao, D.-P. Fan, M.-M. Cheng, X.-Y. Li, and L. Zhang, "Contrast prior and fluid pyramid integration for rgbd salient object detection," in *CVPR*, 2019.

[4] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *TPAMI*, 2017.

[5] N. Liu and J. Han, "Dhsnet: Deep hierarchical saliency network for salient object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 678–686, 2016.

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

[7] C. Lang, T. V. Nguyen, H. Katti, K. Yadati, M. Kankanhalli, and S. Yan, "Depth matters: Influence of depth cues on visual saliency," in *ECCV*, 2012.

[8] D. E. Jacobs, D. B. Goldman, and E. Shechtman, "Cosaliency: Where people look when comparing images," in *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pp. 219–228, 2010.

[9] W. Wang, Q. Lai, H. Fu, J. Shen, H. Ling, and R. Yang, "Salient object detection in the deep learning era: An in-depth survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 6, pp. 3239–3259, 2021.

[10] X. Zhao, L. Zhang, Y. Pang, H. Lu, and L. Zhang, "A single stream network for robust and real-time rgb-d salient object detection," in *ECCV*, 2020.

[11] Z. Zhang, W. Jin, J. Xu, and M.-M. Cheng, "Gradient-induced co-saliency detection," in *ECCV*, 2020.

[12] T. Tieleman and G. Hinton, "Rmsprop: Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning," *COURSERA Neural Networks Mach. Learn*, vol. 17, 2012.

[13] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[14] L. N. Smith, "A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay," *arXiv preprint arXiv:1803.09820*, 2018.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[16] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *CVPR*, 2017.

[17] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár, "Learning to refine object segments," in *European conference on computer vision*, pp. 75–91, Springer, 2016.

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

[19] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[20] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2881–2890, 2017.

[21] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[22] Q. Yan, L. Xu, J. Shi, and J. Jia, "Hierarchical saliency detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1155–1162, 2013.

[23] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, "Saliency detection via graph-based manifold ranking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3166–3173, 2013.

[24] L. Wang, H. Lu, Y. Wang, M. Feng, D. Wang, B. Yin, and X. Ruan, "Learning to detect salient objects with image-level supervision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 136–145, 2017.

[25] G. Li and Y. Yu, "Visual saliency based on multiscale deep features," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5455–5463, 2015.

[26] V. Movahedi and J. H. Elder, "Design and perceptual validation of performance measures for salient object segmentation," in *2010 IEEE computer society conference on computer vision and pattern recognition-workshops*, pp. 49–56, IEEE, 2010.

[27] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.

[28] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," in *2010 IEEE computer society conference on computer vision and pattern recognition*, pp. 3485–3492, IEEE, 2010.

[29] R. Ju, Y. Liu, T. Ren, L. Ge, and G. Wu, "Depth-aware salient object detection using anisotropic center-surround difference," *Signal Processing: Image Communication*, 2015.

[30] H. Peng, B. Li, W. Xiong, W. Hu, and R. Ji, "Rgbd salient object detection: a benchmark and algorithms," in *ECCV*, 2014.

[31] N. Li, J. Ye, Y. Ji, H. Ling, and J. Yu, "Saliency detection on light field," in *CVPR*, 2014.

[32] Y. Cheng, H. Fu, X. Wei, J. Xiao, and X. Cao, "Depth enhanced saliency detection method," in *Conference on Internet Multimedia Computing and Service*, 2014.

[33] D. Fan, Z. Lin, J. Zhao, Y. Liu, Z. Zhang, Q. Hou, M. Zhu, and M. Cheng, "Rethinking RGB-D salient object detection: Models, datasets, and large-scale benchmarks," *CoRR*, vol. abs/1907.06781, 2019.

[34] Y. Piao, W. Ji, J. Li, M. Zhang, and H. Lu, "Depth-induced multi-scale recurrent attention network for saliency detection," in *ICCV*, 2019.

[35] D.-P. Fan, Z. Lin, G.-P. Ji, D. Zhang, H. Fu, and M.-M. Cheng, "Taking a deeper look at co-salient object detection," in *CVPR*, 2020.

[36] D. Zhang, J. Han, C. Li, J. Wang, and X. Li, "Detection of co-salient objects by looking deep and wide," *IJCV*, 2016.

[37] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[38] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari, "Learning object class detectors from weakly annotated video," in *2012 IEEE Conference on computer vision and pattern recognition*, pp. 3282–3289, IEEE, 2012.

[39] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *CVPR*, 2009.

[40] F. Perazzi, P. Krähenbühl, Y. Pritch, and A. Hornung, "Saliency filters: Contrast based filtering for salient region detection," in *2012 IEEE conference on computer vision and pattern recognition*, pp. 733–740, IEEE, 2012.

[41] D.-P. Fan, M.-M. Cheng, Y. Liu, T. Li, and A. Borji, "Structure-measure: A new way to evaluate foreground maps," in *Proceedings of the IEEE international conference on computer vision*, pp. 4548–4557, 2017.

[42] D.-P. Fan, Y. C. Cheng Gong, M.-M. C. Bo Ren, and A. Borji, "Enhanced-alignment measure for binary foreground map evaluation," in *IJCAI*, 2018.

[43] Z. Ren, S. Gao, L.-T. Chia, and I. W.-H. Tsang, "Region-based saliency detection and its application in object recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 5, pp. 769–779, 2013.

[44] D. Zhang, D. Meng, L. Zhao, and J. Han, "Bridging saliency detection to weakly supervised object detection based on self-paced curriculum learning," *arXiv preprint arXiv:1703.01290*, 2017.

[45] T. Liu, J.-J. Huang, T. Dai, G. Ren, and T. Stathaki, "Gated multi-layer convolutional feature extraction network for robust pedestrian detection," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3867–3871, IEEE, 2020.

[46] S. Hong, T. You, S. Kwak, and B. Han, "Online tracking by learning discriminative saliency map with convolutional neural network," in *International conference on machine learning*, pp. 597–606, PMLR, 2015.

[47] X. Yuan, J. Yue, and Y. Zhang, "Rgb-d saliency detection: Dataset and algorithm for robot vision," in *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1028–1033, IEEE, 2018.

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

[49] Y. Wei, J. Feng, X. Liang, M.-M. Cheng, Y. Zhao, and S. Yan, "Object region mining with adversarial erasing: A simple classification to semantic segmentation approach," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1568–1576, 2017.

[50] X. Wang, S. You, X. Li, and H. Ma, "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, 2018.

[51] Y.-F. Ma, L. Lu, H.-J. Zhang, and M. Li, "A user attention model for video summarization," in *Proceedings of the tenth ACM international conference on Multimedia*, pp. 533–542, 2002.

[52] D. Simakov, Y. Caspi, E. Shechtman, and M. Irani, "Summarizing visual data using bidirectional similarity," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, IEEE, 2008.

[53] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[54] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1, pp. 886–893, Ieee, 2005.

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

[56] W. Wang, S. Zhao, J. Shen, S. C. Hoi, and A. Borji, "Salient object detection with pyramid attention and salient edges," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1448–1457, 2019.

[57] X. Qin, Z. Zhang, C. Huang, C. Gao, M. Dehghan, and M. Jagersand, "Basnet: Boundary-aware salient object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7479–7489, 2019.

[58] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[59] I. Bello, B. Zoph, A. Vaswani, J. Shlens, and Q. V. Le, "Attention augmented convolutional networks," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3286–3295, 2019.

[60] P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens, "Stand-alone self-attention in vision models," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[61] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, 2018.

[62] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, and T.-S. Chua, "Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5659–5667, 2017.

[63] T. Zhao and X. Wu, "Pyramid feature attention network for saliency detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3085–3094, 2019.

[64] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. H. Torr, "Deeply supervised salient object detection with short connections," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3203–3212, 2017.

[65] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Icml*, 2010.

[66] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[67] L. Wang, H. Lu, X. Ruan, and M.-H. Yang, "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, 2015.

[68] P. Zhang, D. Wang, H. Lu, H. Wang, and B. Yin, "Learning uncertain convolutional features for accurate saliency detection," in *Proceedings of the IEEE International Conference on computer vision*, pp. 212–221, 2017.

[69] P. Zhang, D. Wang, H. Lu, H. Wang, and X. Ruan, "Amulet: Aggregating multi-level convolutional features for salient object detection," in *Proceedings of the IEEE international conference on computer vision*, pp. 202–211, 2017.

[70] Z. Deng, X. Hu, L. Zhu, X. Xu, J. Qin, G. Han, and P.-A. Heng, "R3net: Recurrent residual refinement network for saliency detection," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pp. 684–690, AAAI Press Menlo Park, CA, USA, 2018.

[71] N. Liu, J. Han, and M.-H. Yang, "Picanet: Learning pixel-wise contextual attention for saliency detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3089–3098, 2018.

[72] T. Wang, L. Zhang, S. Wang, H. Lu, G. Yang, X. Ruan, and A. Borji, "Detect globally, refine locally: A novel approach to saliency detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3127–3135, 2018.

[73] L. Zhang, J. Dai, H. Lu, Y. He, and G. Wang, "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, 2018.

[74] R. Wu, M. Feng, W. Guan, D. Wang, H. Lu, and E. Ding, "A mutual learning method for salient object detection with intertwined multi-supervision," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8150–8159, 2019.

[75] M. Feng, H. Lu, and E. Ding, "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, 2019.

[76] A. Siris, J. Jiao, G. K. Tam, X. Xie, and R. W. Lau, "Scene context-aware salient object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4156–4166, 2021.

[77] Y.-C. Gu, S.-H. Gao, X.-S. Cao, P. Du, S.-P. Lu, and M.-M. Cheng, "Inas: integral nas for device-aware salient object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4934–4944, 2021.

[78] L. Tang, B. Li, Y. Zhong, S. Ding, and M. Song, "Disentangled high quality salient object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3580–3590, 2021.

[79] A. Li, J. Zhang, Y. Lv, B. Liu, T. Zhang, and Y. Dai, "Uncertainty-aware joint salient object and camouflaged object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10071–10081, 2021.

[80] H. Peng, B. Li, W. Xiong, W. Hu, and R. Ji, "Rgbd salient object detection: a benchmark and algorithms," in *ECCV*, 2014.

[81] J. Ren, X. Gong, L. Yu, W. Zhou, and M. Ying Yang, "Exploiting global priors for rgb-d saliency detection," in *CVPRW*, 2015.

[82] H. Song, Z. Liu, H. Du, G. Sun, O. Le Meur, and T. Ren, "Depth-aware salient object detection and segmentation via multiscale discriminative saliency fusion and bootstrap learning," *TIP*, 2017.

[83] Z. Liu, S. Shi, Q. Duan, W. Zhang, and P. Zhao, "Salient object detection for rgb-d image by single stream recurrent convolution neural network," *Neurocomputing*, 2019.

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

[85] T. Zhou, D.-P. Fan, M.-M. Cheng, J. Shen, and L. Shao, "Rgb-d salient object detection: A survey," *Computational Visual Media*, vol. 7, no. 1, pp. 37–69, 2021.

[86] R. Cong, J. Lei, C. Zhang, Q. Huang, X. Cao, and C. Hou, "Saliency detection for stereoscopic images based on depth confidence analysis and multiple cues fusion," *IEEE Signal Processing Letters*, 2016.

[87] J. Guo, T. Ren, and J. Bei, "Salient object detection for rgb-d image via saliency evolution," in *ICME*, 2016.

[88] A. Ciptadi, T. Hermans, and J. M. Rehg, "An in depth view of saliency," 2013.

[89] Y. Pang, L. Zhang, X. Zhao, and H. Lu, "Hierarchical dynamic filtering network for rgb-d salient object detection," *arXiv preprint arXiv:2007.06227*, 2020.

[90] S. Chen and Y. Fu, "Progressively guided alternate refinement network for rgb-d salient object detection," in *ECCV*, 2020.

[91] Y. Piao, Z. Rong, M. Zhang, W. Ren, and H. Lu, "A2dele: Adaptive and attentive depth distiller for efficient rgb-d salient object detection," in *CVPR*, 2020.

[92] F. Huo, X. Zhu, L. Zhang, Q. Liu, and Y. Shu, "Efficient context-guided stacked refinement network for rgb-t salient object detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 5, pp. 3111–3124, 2021.

[93] Q. Duan, L. Zhang, and X. Gao, "Simultaneous face completion and frontalization via mask guided two-stage gan," *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.

[94] Y. Pang, J. Xie, M. H. Khan, R. M. Anwer, F. S. Khan, and L. Shao, "Mask-guided attention network for occluded pedestrian detection," in *ICCV*, 2019.

[95] W. Liu, A. Rabinovich, and A. C. Berg, "Parsenet: Looking wider to see better," *arXiv preprint arXiv:1506.04579*, 2015.

[96] C. Zhu, G. Li, W. Wang, and R. Wang, "An innovative salient object detection using center-dark channel prior," in *ICCVW*, 2017.

[97] L. Qu, S. He, J. Zhang, J. Tian, Y. Tang, and Q. Yang, "Rgbd salient object detection via deep fusion," *TIP*, 2017.

[98] N. Wang and X. Gong, "Adaptive fusion for rgb-d salient object detection," *IEEE Access*, 2019.

[99] H. Chen, Y. Li, and D. Su, "Multi-modal fusion network with multi-scale multi-path and cross-modal interactions for rgb-d salient object detection," *Pattern Recognition*, 2019.

[100] H. Chen and Y. Li, "Three-stream attention-aware network for rgb-d salient object detection," *TIP*, 2019.

[101] J. Han, H. Chen, N. Liu, C. Yan, and X. Li, "Cnns-based rgb-d saliency detection via cross-view transfer and multiview fusion," *IEEE Transactions on Cybernetics*, 2017.

[102] W. Luo, Y. Li, R. Urtasun, and R. Zemel, "Understanding the effective receptive field in deep convolutional neural networks," *arXiv preprint arXiv:1701.04128*, 2017.

[103] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[104] P. Zhang, L. Su, L. Li, B. Bao, P. Cosman, G. Li, and Q. Huang, "Training efficient saliency prediction models with knowledge distillation," in *ACM*, 2019.

[105] X. Cheng, Z. Rao, Y. Chen, and Q. Zhang, "Explaining knowledge distillation by quantifying the knowledge," in *CVPR*, 2020.

[106] S. Liu, D. Huang, *et al.*, "Receptive field block net for accurate and fast object detection," in *ECCV*, 2018.

[107] X. Zhou, H. Wen, R. Shi, H. Yin, J. Zhang, and C. Yan, "Fanet: Feature aggregation network for rgbd saliency detection," *Signal Processing: Image Communication*, vol. 102, p. 116591, 2022.

[108] G. Ren, Y. Xie, T. Dai, and T. Stathaki, "Progressive multi-scale fusion network for rgb-d salient object detection," *Computer Vision and Image Understanding*, vol. 223, p. 103529, 2022.

[109] J. Zhang, D.-P. Fan, Y. Dai, X. Yu, Y. Zhong, N. Barnes, and L. Shao, "Rgb-d saliency detection via cascaded mutual information minimization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4338–4347, 2021.

[110] G. Liu and D. Fan, "A model of visual attention for natural image retrieval," in *ISCC-C*, 2013.

[111] Z. Gao, C. Xu, H. Zhang, S. Li, and V. H. C. de Albuquerque, "Trustful internet of surveillance things based on deeply represented visual co-saliency detection," *IEEE IoT-J*, 2020.

[112] K.-J. Hsu, Y.-Y. Lin, and Y.-Y. Chuang, "Deepco3: Deep instance co-segmentation by co-peak search and co-saliency detection," in *CVPR*, 2019.

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

[114] L. Wei, S. Zhao, O. E. F. Bourahla, X. Li, and F. Wu, "Group-wise deep co-saliency detection," *arXiv preprint arXiv:1707.07381*, 2017.

[115] K. Zhang, T. Li, S. Shen, B. Liu, J. Chen, and Q. Liu, "Adaptive graph convolutional network with attention graph clustering for co-saliency detection," in *CVPR*, 2020.

[116] Z.-J. Zha, C. Wang, D. Liu, H. Xie, and Y. Zhang, "Robust deep co-saliency detection with group semantic and pyramid attention," *IEEE TNNLS*, 2020.

[117] B. Li, Z. Sun, Q. Li, Y. Wu, and A. Hu, "Group-wise deep object co-segmentation with co-attention recurrent neural network," in *ICCV*, 2019.

[118] B. Mahasseni, M. Lam, and S. Todorovic, "Unsupervised video summarization with adversarial lstm networks," in *CVPR*, 2017.

[119] Y. Wang, X. Zhang, Z. Liu, Z. Dong, X. Feng, R. Tang, and X. He, "Personalized re-ranking for improving diversity in live recommender systems," *arXiv preprint arXiv:2004.06390*, 2020.

[120] T. Dai, H. Liu, K. Arulkumaran, G. Ren, and A. A. Bharath, "Diversity-based trajectory and goal selection with hindsight experience replay," in *PRICAI*, 2021.

[121] T. Dai, Y. Du, M. Fang, and A. A. Bharath, "Diversity-augmented intrinsic motivation for deep reinforcement learning," *Neurocomputing*, 2022.

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

[123] H. Fu, X. Cao, and Z. Tu, "Cluster-based co-saliency detection," *IEEE TIP*, 2013.

[124] B. Li, Z. Sun, L. Tang, Y. Sun, and J. Shi, "Detecting robust co-saliency with recurrent co-attention neural network.," in *IJCAI*, 2019.

[125] Y. Li, K. Fu, Z. Liu, and J. Yang, "Efficient saliency-model-guided visual co-saliency detection," *IEEE SPL*, 2014.

[126] J. Han, G. Cheng, Z. Li, and D. Zhang, "A unified metric learning-based framework for co-saliency detection," *IEEE TCSVT*, 2017.

[127] K. Zhang, T. Li, B. Liu, and Q. Liu, "Co-saliency detection via mask-guided fully convolutional networks with multi-scale label smoothing," in *CVPR*, 2019.

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

[129] Z. Wu, L. Su, and Q. Huang, "Stacked cross refinement network for edge-aware salient object detection," in *ICCV*, 2019.