Saliency Hierarchy Modeling via Generative Kernels for Salient Object Detection

Wenhu Zhang[1], Liangli Zheng[2], Huanyu Wang[3], Xintian Wu[3], and Xi Li[⋆3,4,5]

[1] Polytechnic Institute, Zhejiang University
[2] School of Software Technology, Zhejiang University
[3] College of Computer Science and Technology, Zhejiang University
[4] Shanghai Institute for Advanced Study, Zhejiang University
[5] Shanghai Al Laboratory
{wenhuzhang, lianglizheng, huanyuhello, hsintien, xilizju}@zju.edu.cn

Abstract. Salient Object Detection (SOD) is a challenging problem that aims to precisely recognize and segment the salient objects. In ground-truth maps, all pixels belonging to the salient objects are positively annotated with the same value. However, the saliency level should be a relative quantity, which varies among different regions in a given sample and different samples. The conflict between various saliency levels and single saliency value in ground-truth, results in learning difficulty. To alleviate the problem, we propose a Saliency Hierarchy Network (SHNet), modeling saliency patterns via generative kernels from two perspectives: region-level and sample-level. Specifically, we construct a Saliency Hierarchy Module to explicitly model saliency levels of different regions in a given sample with the guide of prior knowledge. Moreover, considering the sample-level divergence, we introduce a Hyper Kernel Generator to capture the global contexts and adaptively generate convolution kernels for various inputs. As a result, extensive experiments on five standard benchmarks demonstrate our SHNet outperforms other state-of-the-art methods in both terms of performance and efficiency.

Keywords: Salient object detection, Saliency hierarchy modeling, Region-level, Sample-level, Generative kernel

1 Introduction

Salient Object Detection (SOD) aims to accurately detect and segment the most eye-catching area in a given image, mimicking the human visual perception. In recent years, deep learning based SOD methods have achieved huge success by introducing dense feature interactions [6, 27, 45], various attention modules [11, 36, 54], and multi-task learning pipelines [43, 51, 53]. In essence, these approaches leverage the strong abilities of the deep neural networks to learn a mapping function from raw images to ground-truth saliency maps, in which the whole salient object is positively annotated with the same saliency level.

⋆ Corresponding author

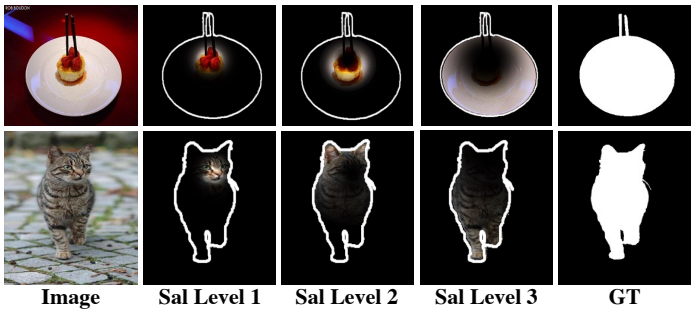

Fig. 1: The illustration of saliency hierarchy. Although all pixels in the salient objects are positively annotated with the same value in ground-truth maps (GT), the saliency levels (Sal Level) of different regions are inherently hierarchical.

However, it shows evident saliency divergence among different regions (in a given sample) and samples, due to various colors, sizes, layouts, etc. For example, in the first row of Fig. 1, the fruits, cake and plate obviously possess diverse saliency levels. Similarly, in the second row, different regions inside the cat also show different saliency levels. It is difficult to learn a mapping function from pixels with divergent saliency levels to the same value in a ground-truth map. To solve this problem, we propose a novel framework modeling the hierarchical saliency levels with generative kernels for different regions and samples.

In order to model the region-level hierarchical saliency, a straightforward solution is to learn annotated hierarchical labels. As a matter of fact, there is no acknowledged standard for saliency level division in the literature. Therefore, we explore several saliency level decomposition strategies and generate sub-saliency masks of different saliency levels. Based on these sub-saliency masks, we design a Saliency Hierarchy Module (SHM). Firstly, the SHM extracts local features within regions of different saliency levels. An SHM contains multiple branches and regions of features belonging to the same saliency level are processed in an independent branch. Then, it aggregates all the region level features together forming an integrated feature map. Through cascading several SHMs together, we achieve a hierarchy saliency modeling scheme at every stage of our framework. In this way, we not only depict the saliency levels explicitly, but also model hierarchical saliency levels with different patterns.

Moreover, considering the sample-level saliency divergence, we propose to adaptively generate convolution kernels for different samples. We design a Hyper Kernel Generator (HKG) to capture a global view of the input image. Specifically, we introduce a Transformer decoder [38] to generate a set of hyper-kernels by constructing dense attention between several learnable queries and flattened image patches. Through sufficient interactions, the hyper-knowledge of different saliency patterns are embedded in the hyper-kernels. Each hyper-kernel corresponds to a specific saliency level in SHMs. With these hyper-kernels, HKG further produces different convolution kernels for all branches in SHMs. Differ-

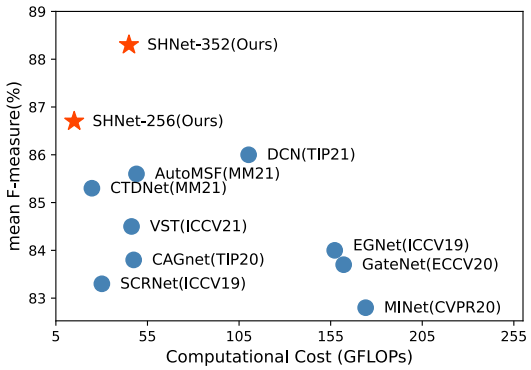

Fig. 2: Mean F-measure against GFLOPs on DUTS-TE [40]. Our SHNet (Red Stars) outperforms other SOTA approaches on both performance and efficiency.

ent from previous works [4, 7], who utilize transformer to predict task-specific elements, our HKG aims to generate convolutional kernels for the decoder, improving the capacity and flexibility of our framework. As a result, we generate convolutional kernels to model saliency divergence among different samples with the proposed HKG, and model region-level saliency with SHMs. Our framework achieves the state-of-the-art performance, as shown in Fig. 2.

The main contributions are summarized as follows:

- We propose a novel framework to model salient objects hierarchically from the perspective of regions and samples.
- We design Saliency Hierarchy Modules (SHM), which model the region-level saliency hierarchy within a given sample.
- We introduce a Hyper Kernel Generator (HKG) to generate adaptive kernels to model the saliency divergence among samples.
- Extensive experiments on five widely used benchmarks demonstrate our method achieves SOTA results *w.r.t.*, performance and efficiency.

2 Related Work

2.1 Salient Object Detection

Early SOD methods [1, 2, 15, 18, 19, 26, 41] mainly focus on the hand-crafted features to detect and segment the salient objects, such as color contrast [2], frequency prior [1], etc. Recently, deep learning based SOD approaches [5, 16, 32,48] have achieved a qualitative leap in performance due to the powerful feature extraction capability in visual representation. The existing SOD approaches can be roughly divided into architecture based methods [12, 25, 27, 30, 36, 39, 42, 56] and regularization based methods [21, 24, 37, 43, 47, 48].

Architecture based methods. The architecture based methods mainly concentrate on designing novel models for the complex feature interaction. For example, Pang *et al.* [30] used mutual learning and the self-interaction module to reduce the noise during feature fusion. Wei *et al.*(F3Net) [42] used the cascaded feedback decoder to release the feature redundancy between various levels. Ma *et al.* [27] proposed a pyramidal feature shrinking network to aggregate adjacent feature nodes in pairs and discard interference information. Other methods [12, 25, 36, 39, 56] also verify the effectiveness of dense interaction for SOD.

Regularization based methods. As for the regularization based approaches, they improve the performance by building auxiliary supervision. For example, Wei *et al.* [43] (LDF) used the body map and detail map as auxiliary supervision to avoid the interference between the center area and boundary. Liu *et al.* [24] introduced edge detection and skeleton extraction, trying to solve them with SOD jointly. Tang *et al.* [37] designed an uncertainty-based saliency map to disentangle high-resolution SOD into two tasks and achieve good results. With similar purpose, various auxiliary loss functions [21, 47, 48] are proposed for regularizing the training of deep SOD models.

Different from these methods, which directly map the whole images to corresponding binary saliency maps, we focus on hierarchically modeling saliency patterns and alleviating the learning difficulty caused by the saliency divergence.

2.2 Hypernetworks and Transformers.

Hypernetworks. Hypernetwork [13, 17] can directly generate instance-wise parameters for the network with another independent weight generation model at test time. It is a powerful modeling tool providing the network adaptivity in a parameter-efficient manner. Similar design is used for many tasks, such as image-to-image translation [17], semantic segmentation [29], 3D reconstruction [23], and so on. Inspired by hypernetworks, we utilize a shared HKG module for generating sample-adaptive kernels for all the HSMs, improving the model capacity with a computational-efficient manner.

Transformers. Vaswani *et al.* [38] proposed the first transformer encoder-decoder architecture for NLP tasks. Recently, various computer vision tasks introduce transformer models and achieve exciting results, including image classification [9], semantic segmentation [57], object detection [4] and saliency object detection [25]. Different from CNN-based models, transformer relies on the attention mechanism to model the long-term dependencies from a sequence perspective. VST [25] first introduces the transformer architecture to SOD task for capturing the global contexts and contrast, which achieves huge success. In this paper, our HKG can be treated as a meta block or hypernetwork, which aims to generate unique convolution kernels for per image and per saliency level, modeling the sample-level saliency divergence adaptively.

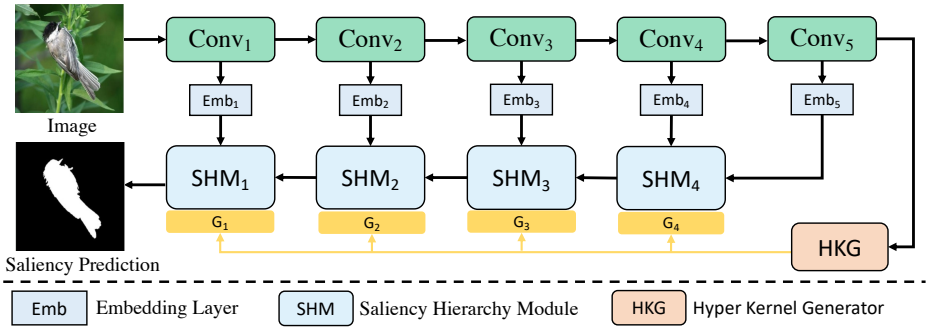

Fig. 3: An overview of our proposed Saliency Hierarchy Network. The whole network consists of a CNN backbone with embedding layers, a Hyper Kernel Generator and five stacked Saliency Hierarchy Modules.

3 Method

In this section, we illustrate our Saliency Hierarchy Network (SHNet) in detail. The proposed SHNet mainly consists of two modules, Saliency Hierarchy Module (SHM) and Hyper Kernel Generator (HKG), as shown in Fig. 3. Within an SHM, we produce several sub-saliency masks through a saliency level classifier and utilize these masks to decompose the input features into several parts for hierarchical saliency modeling in respective branches. Note that the sub-saliency masks are regularized by prior knowledge to depict the hierarchy decomposition process explicitly. Within an HKG, we utilize a transformer decoder to generate sample-adaptive hyper-kernels, which are further parsed into convolution kernel groups corresponding to all the cascaded SHMs. We utilize the traditional CNN backbone (e.g., ResNet [14], VGG [35]) as the encoder, containing K-layer convolution blocks. Formally, we denote the number of saliency levels as N and the outputs of each encoder layers as $\{F_1, F_2, ..., F_K\}$.

3.1 Saliency Hierarchy Module

In this section, we describe the detail of Saliency Hierarchy Module in detail. The SHM focuses on region-level hierarchical saliency modeling within a given sample. As shown in Fig. 4, an SHM includes two processes: Saliency Feature Decomposition and Hierarchical Saliency Modeling.

Saliency Feature Decomposition. In order to decompose the input features according to their different saliency levels, we propose to depict the saliency hierarchy with generated sub-saliency masks. Specifically, we introduce an N-class classifier to predict the pixel-wise saliency levels:

$$\hat{P}_k = \text{softmax}(\text{Conv}_{3\times3}(H_k)), \tag{1}$$

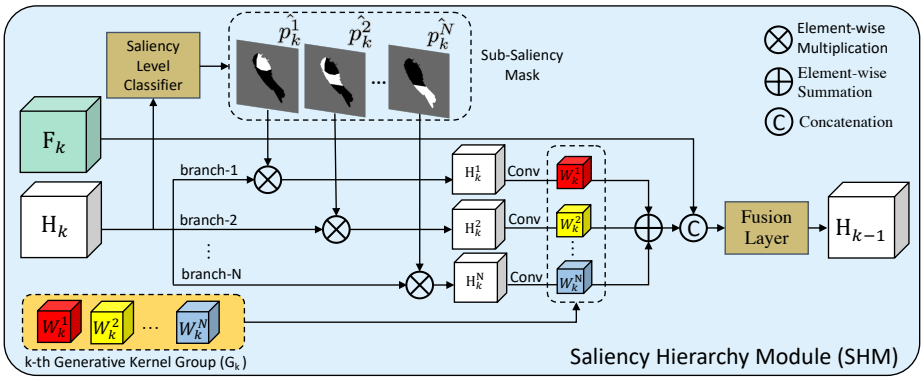

Fig. 4: The detailed depiction of our proposed Saliency Hierarchy Module.

where H_k is the input features of k-th SHM, $\hat{P}_k$ is the stacked sub-saliency masks, softmax$(\cdot)$ is the softmax normalization along the channel dimension, and $\text{Conv}_{3\times3}(\cdot)$ is a learnable convolution layer. Then, we unfold the obtained $\hat{P}_k$ and get N predicted sub-saliency masks $\{\hat{p_k^1}, ..., \hat{p_k^N}\}$, corresponding to N different levels in the salient objects.

Moreover, to assign different saliency patterns to various regions, we decompose the input features into sub-features corresponding to respective saliency levels under the guidance of the obtained sub-saliency masks:

$$H_k^n = (\hat{p_k^n} \otimes H_k), \quad n = 1, 2, ..., N, \tag{2}$$

where $\otimes$ denotes the element-wise multiplication and H_k^n is the obtained sub-feature for the n-th saliency level.

It has to be mentioned that several prior knowledge is introduced for providing pseudo-labels $\{p_k^1, ..., p_k^N\}$ to regularize the predicted sub-saliency masks. During the regularization, we explicitly guide the learning process of the sub-saliency masks and inject the prior knowledge into our framework to mimic the saliency hierarchy.

Hierarchical Saliency Modeling. In order to detect salient objects in different levels, we input the decomposed features $\{H_k^1, ..., H_k^N\}$ in multiple branches. Specifically, we use the different convolution kernels to extract features from different levels, achieving more dedicated saliency modeling schemes as:

$$H_k^{n\prime} = H_k^n * W_k^n, \quad n = 1, 2, ..., N, \tag{3}$$

where W_k^n is the n-th kernel of the input Generative Kernel Groups, $*$ is the convolution operation, and $H_k^{n\prime}$ indicates the features from different regions. Finally, we aggregate all regional features $\{H_k^{1\prime}, ..., H_k^{N\prime}\}$ with the input feature

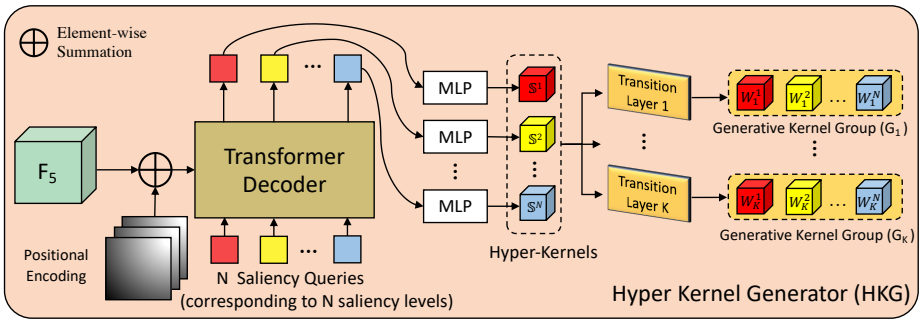

Fig. 5: The detailed depiction of our proposed Hyper Kernel Generator.

F_k from the k-th CNN block. The whole process can be formulated as:

$$H_{k-1} = \text{Conv}_{3\times3}(\text{Concat}(\sum_{n=1}^{N} H_k^{n\prime}, F_k)), \tag{4}$$

where $\text{Concat}(\cdot)$ is the concatenation operator and H_{k-1} is the output of SHM_k.

Overall, SHM not only explicitly depicts the hierarchical saliency levels inside an image based on sub-saliency masks, but also models the saliency patterns of different saliency levels to capture more details within respective branches.

3.2 Hyper Kernel Generator

In this section, we illustrate the proposed Hyper Kernel Generator (HKG), which further promotes our hierarchy saliency modeling scheme adaptive to different samples, as shown in Fig. 5. Inspired by hypernetworks, we utilize a unified HKG to generate the cascaded Generative Kernel Groups for all the HSM modules. First, the shared hyper-kernels are produced by a transformer block. Each hyper-kernel corresponds to a saliency level. Next, these shared hyper-kernels are parsed into different kernel groups prepared for the cascaded SHMs.

Hyper-Kernels. In order to get the hyper-kernels, we introduce transformer architecture to establish dense attention between several learnable queries and flattened image patches. The transformer block is composed of L stacked transformer decoding layers. Each layer of the transformer constructs interactions between the learnable saliency queries Q_0 and the flattened image patches. In this way, the l-th transformer decoding layer is formulated as:

$$Q_l = \text{MLP}(\text{MCA}(\text{MSA}(Q_{l-1}), T)), \tag{5}$$

where $\text{MSA}(\cdot)$ is the multi-head self-attention, $\text{MCA}(\cdot)$ is the multi-head cross-attention, and $\text{MLP}(\cdot)$ is the multi-layer perceptron blocks. T stands for the flattened input F_K with standard positional encoding. Denote Q_L as the output

of the last transformer layer. We use a shared MLP to project the output feature Q_L into hyper-kernels:

$$\mathbb{S}^n = \text{MLP}(Q_L^n), \quad n = 1, 2, ..., N, \tag{6}$$

where Q_L^n is the n-th token in Q_L and $\mathbb{S}^n$ is the hyper-kernel for the n-th saliency level. By utilizing transformer, we achieve a group of sample-adaptive hyper-kernels $\{\mathbb{S}^1, \mathbb{S}^2, ..., \mathbb{S}^N\}$ for a given sample.

Transition Layers. After that, we assign K transition layers parsing the generated hyper-kernels, i.e., $\mathbb{S}^n$, into K different kernel groups:

$$W_k^n = \phi_k(\mathbb{S}^n), \quad k = 1, 2, ..., K, n = 1, 2, ..., N, \tag{7}$$

where $\phi_k(\cdot)$ is the k-th transition layer. Each transition layer is prepared for a specific decoder block (i.e., SHM). Finally, the k-th Generative Kernel Group $G_k = \{W_k^1, W_k^2, ..., W_k^N\}$ is fed into SHM_k to conduct the sample-adaptive convolution operations. It is worth noting that the hyper-kernels are shared across the all SHMs.

Overall, our HKG module learns the hyper-knowledge of diverse saliency patterns to generate the cascaded sample-adaptive kernel groups for decoder blocks, enhancing the flexibility and capacity of the framework dramatically.

3.3 Optimization

Sub-saliency Masks Regularization. As a matter of fact, it is a daunting task to generate sub-saliency masks in a completely unconstrained situation. To alleviate the learning difficulty, we introduce a prior guidance G_{sal} to divide the salient objects of a ground-truth label into several parts $\{p^1, p^2, ..., p^N\}$. p^n denotes the sub-saliency label of the n-th saliency level, and all pixels belong to the n-th saliency level are annotated as positive, otherwise negative. For example, we use Grad-CAM [34] to obtain the gradient response map of the input sample and divide the ground-truth label according to the level of the gradient response. More prior algorithms and corresponding experiments are discussed in Section 4.4.

Objective Function. Based on the obtained sub-saliency labels $\{p^1, p^2, ..., p^N\}$, we propose to regularize the predicted sub-saliency masks $\{\hat{p^1}, \hat{p^2}, ..., \hat{p^N}\}$. Note that we only regularize the pixels that belong to the salient objects y_{pos}, i.e., white and black regions in the sub-saliency mask $\hat{p^n}$ in Fig. 4. Those pixels in gray are ignored. Thus, the saliency hierarchy loss for n-th saliency level in the k-th SHM is denoted as:

$$\mathcal{L}_{hierarchy}^{n,k} = \sum_{(i,j) \in y_{pos}} (\hat{p_k^n}(i,j) - p^n(i,j))^2, \tag{8}$$

where $\hat{p}_k^n(i,j)$ and $p^n(i,j)$ are pixels at location (i,j) from the predicted sub-saliency mask and sub-saliency label, respectively.

Finally, the overall objective function is the combination between $\mathcal{L}_{hierarchy}^{n,k}$ and a pixel position aware loss L_{ppa} [42], denoted as:

$$\mathcal{L}_{total} = \mathcal{L}_{ppa}(\hat{y}, y) + \rho \sum_{k=1}^{K} \sum_{n=1}^{N} \mathcal{L}_{hierarchy}^{n,k}, \tag{9}$$

where ρ is a hyper-parameter, $\hat{y}$ and y are the predicted and ground-truth saliency maps, respectively.

4 Experiments

4.1 Datasets and Evaluation Metrics

Datasets. We perform experiments on five widely used benchmark datasets, including DUTS [40], ECSSD [49], HKU-IS [20], DUT-O [50] and PASCAL-S [22]. DUTS contains 10,553 training images (DUTS-TR) and 5,019 test images (DUTS-TE). ECSSD contains 1000 images with structurally complex natural contents. HKU-IS is composed of 4,447 complex scenes that contain multiple salient objects. DUT-O contains 5,168 images with complex backgrounds. PASCAL-S consists of 850 challenging images.

Evaluation Metrics. To comprehensively and fairly evaluate various methods, we employ three widely used metrics, including mean F-measure ($\mathcal{F}_\beta$) [1], mean absolute error ($\mathcal{M}$) [3] and E-measure ($\mathcal{E}_\xi$) [10]. Specifically, the mean F-measure can evaluate the overall performance based on the region similarity. The Mean Absolute Error represents the average absolute difference between the saliency map and ground truth. The E-measure can jointly utilize image-level statistics and local pixel-level statistics for evaluating the binary saliency map. Besides, we also report the floating point of operations (FLOPs) to evaluate their complexity.

4.2 Implementation Details

Our method is implemented with PyTorch toolbox [31], and can be conducted on a single NVIDIA GTX 1080Ti GPU. The proposed model is trained on DUTS-TR and tested on the above mentioned five datasets. As for training, we adopt ResNet-50 [14] and VGG-16 [35] as our backbone networks, which are pretrained on the ImageNet [8] dataset. To reduce overfitting, we utilize image augmentation techniques (i.e., random flipping, rotating, cropping and color enhancing). The maximum learning rates are set to 2e-5 for the convolution backbone network and 2e-4 for other parts, with warm-up and linear decay strategies. Training batch size and epochs are set to 16 and 100, respectively. Totally, the whole network is trained in an end-to-end manner using Adam optimizer. For more fair comparison, we conduct experiments on two kinds of resolutions (i.e., 256×256 and 352×352) with different channel settings. As for testing, each image is simply resized to the corresponding resolution and then fed into our network to get the saliency prediction without any post-processing.

Table 1: Performance comparison of SOTA methods over 5 datasets. MAE ($\mathcal{M} \downarrow$, smaller is better), mean F-measure ($F_\beta \uparrow$, larger is better) and mean E-measure ($E_\xi \uparrow$, larger is better) are used to measure the model performance. '†' means inputs of 256 resolution. '∗' means inputs of 352 resolution. The best two results are marked in red and blue. Our method outperforms other approaches on both performance and efficiency.

Method	GFLOPs	DUTS-TE			ECSSD			HKU-IS			DUT-O			PASCAL-S		
		$\mathcal{M}$	F_β	E_ξ	$\mathcal{M}$	F_β	E_ξ	$\mathcal{M}$	F_β	E_ξ	$\mathcal{M}$	F_β	E_ξ	$\mathcal{M}$	F_β	E_ξ
VGG-Based methods																
CPD* [45]	118.86	.043	.813	.902	.040	.915	.938	.033	.896	.940	.057	.745	.845	.074	.825	.882
AFN [11]	-	.046	.812	.893	.042	.905	.935	.036	.888	.934	.057	.742	.846	.071	.824	.883
MLMS* [44]	256.81	.045	.802	.893	.038	.914	.943	.034	.893	.942	.056	.742	.853	.069	.838	.890
EGN† [53]	-	.043	.826	.898	.044	.910	.936	.034	.894	.938	.056	.752	.853	.076	.818	.877
CAGN* [28]	154.25	.044	.823	.904	.042	.915	.939	.033	.906	.947	.057	.744	.860	.077	.831	.881
GateN [55]	216.47	.045	.817	.893	.041	.905	.932	.035	.891	.934	.061	.733	.840	.070	.826	.886
MIN [30]	292.52	.039	.823	.912	.036	.922	.943	.030	.906	.955	.057	.741	.864	.065	.843	.898
ITSD [58]	114.93	.042	.833	.905	.040	.910	.937	.035	.894	.938	.063	.752	.853	.074	.831	.891
DCN* [47]	411.25	.041	-	.918	.032	-	.945	.034	-	.949	.055	-	.871	.069	-	.892
AMSF† [52]	87.71	.039	.842	.920	.036	.924	.951	.029	.908	.955	.056	.763	.866	.068	.840	.899
SHNet†	44.97	.035	.851	.926	.033	.926	.950	.028	.913	.955	.054	.765	.868	.060	.842	.906
SHNet*	95.02	.034	.861	.928	.031	.930	.952	.026	.917	.957	.056	.769	.868	.058	.849	.910
ResNet / Transformer-Based methods																
CPD* [45]	35.48	.043	.805	.898	.037	.917	.942	.034	.891	.938	.056	.747	.847	.072	.824	.882
EGN† [53]	157.21	.039	.839	.907	.041	.918	.943	.031	.902	.944	.052	.760	.857	.074	.823	.881
SCRN* [46]	30.13	.040	.833	.900	.038	.916	.939	.034	.894	.935	.056	.749	.848	.064	.833	.892
CAGN* [28]	47.47	.040	.838	.914	.037	.921	.944	.030	.910	.950	.054	.753	.862	.067	.847	.896
GateN [55]	162.13	.040	.837	.906	.040	.913	.936	.033	.897	.937	.055	.757	.855	.069	.826	.886
F3N* [42]	32.86	.035	.852	.920	.033	.928	.948	.028	.910	.952	.053	.766	.864	.061	.830	.898
MIN [30]	174.06	.037	.828	.917	.033	.924	.953	.028	.908	.956	.055	.756	.873	.064	.842	.899
LDF* [43]	31.02	.034	.855	.910	.034	.930	.925	.027	.914	.954	.051	.773	.873	.060	.848	.865
VST† [25]	46.32	.037	.845	.919	.034	.920	.951	.030	.907	.952	.058	.774	.871	.067	.835	.902
CTDN* [56]	24.66	.034	.853	.929	.032	.927	.950	.027	.919	.955	.052	.779	.875	.061	.841	.901
DCN* [47]	110.22	.035	.860	.927	.032	.931	.955	.027	.915	.958	.051	.779	.878	.062	.839	.901
AMSF† [52]	48.96	.034	.856	.931	.033	.929	.954	.027	.914	.959	.050	.778	.876	.061	.850	.902
SHNet†	15.26	.032	.867	.936	.030	.933	.958	.026	.918	.958	.049	.784	.883	.057	.849	.912
SHNet*	44.87	.030	.883	.938	.028	.939	.957	.025	.926	.959	.048	.790	.880	.056	.855	.910

4.3 Comparison with State-of-the-art

Quantitative Comparison. As shown in Table 1, we present the quantitative comparison in terms of four evaluation metrics on five datasets. On one hand, our SHNet surpasses these SOTA methods by a large margin across all the datasets in most metrics. Specifically, the VGG-based SHNet outperforms other methods across all datasets, except that it ranks second on E_ξ of DUT-O dataset. The ResNet-based SHNet obtains the best results on all five datasets. Especially, the F_β of ResNet-based SHNet-352 is significantly better than other best result on DUTS-TE (88.3% against 86.0%) and DUT-O (79.0% against 77.9%). Meanwhile, it also possesses an evident advantage on $\mathcal{M}$ with 11.7% and 12.5% improvements on DUTS-TE and ECSSD.

On the other hand, our method (SHNet-256) achieves the state-of-the-art performance using the lowest cost (i.e., VGG-based: 44.97 GFLOPs, ResNet-

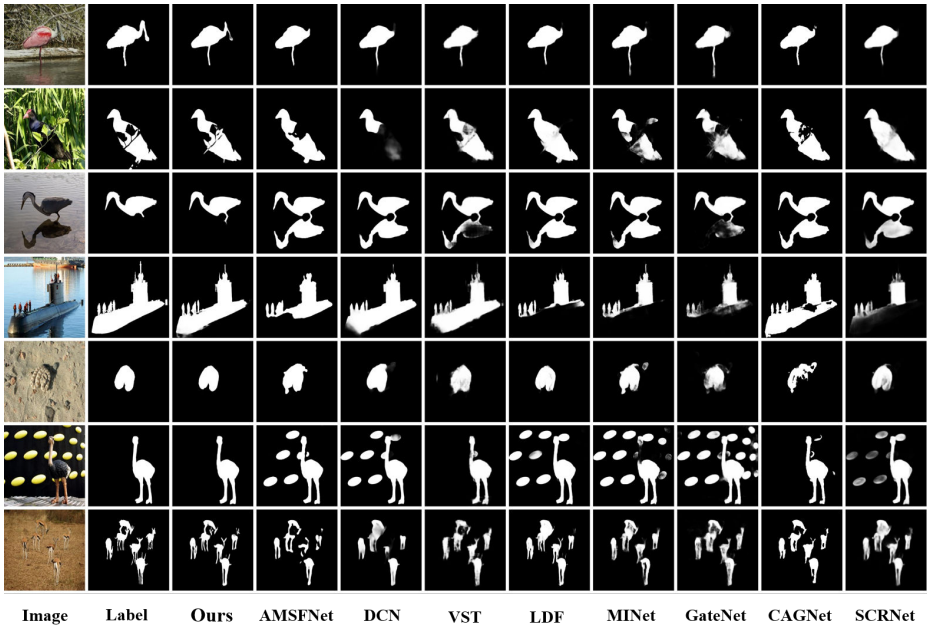

Fig. 6: Qualitative comparison between the state-of-the-art SOD methods and our SHNet. Obviously, saliency maps produced by our model are more clear and more accurate than that of other methods in various challenging scenarios.

based: 15.26 GFLOPs) among all the compared methods. Moreover, both the VGG-based and ResNet-based SHNet-352 achieve higher performance without consuming much computation cost. The reason might be a larger input is helpful to more accurate modeling of region-level saliency divergence.

Qualitative Comparisons. To further illustrate the effectiveness of our method, we visualize a qualitative comparison between our method and other state-of-the-art approaches. As shown in Fig. 6, our method not only highlights the salient object regions clearly, but also well suppresses the background noises. The proposed SHNet is able to accurately segment salient objects under various challenging scenarios, including images with fine structures (1st and 7th rows), partial occlusion (2nd row), reflection interference (3rd row), low contrast foreground and background (4th and 5th rows), and cluttered distractions (6th and 7th rows). It is worth noting that SHNet achieves better results than other methods under an extremely challenging scenario (7th row), where multiple salient objects are similar with the background in terms of color and texture.

4.4 Ablation Analysis

In this section, we perform a series of ablation studies and further estimate each component in the proposed framework. First, we explore different prior guidance

Table 2: Ablation study of different prior guidance G_{sal}. 'w/o' means no extra supervision. 'Baseline' means framework without saliency hierarchy modeling.

#	G_{sal}	DUTS-TE		ECSSD		DUT-O	
		$\mathcal{F}_\beta \uparrow$	$\mathcal{M} \downarrow$	$\mathcal{F}_\beta \uparrow$	$\mathcal{M} \downarrow$	$\mathcal{F}_\beta \uparrow$	$\mathcal{M} \downarrow$
1	w/o	.830	.038	.909	.039	.745	.056
2	Erode	.849	.036	.922	.034	.761	.054
3	DT	.850	.035	.920	.034	.764	.054
4	Grad-Cam	**.854**	**.034**	**.927**	**.031**	**.772**	**.050**
*	Baseline	.820	.041	.901	.041	.734	.060

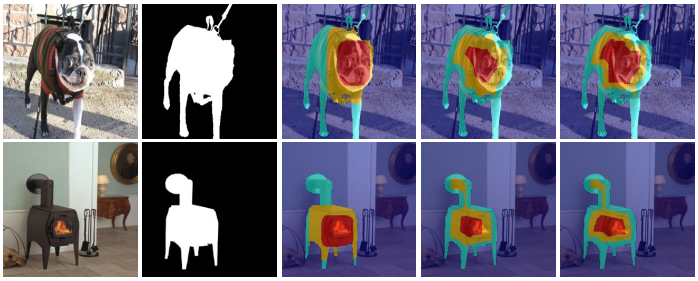

Fig. 7: The visualization results of prior guidance G_{sal}. Red, yellow and green represent the saliency level division results, when the number of saliency levels (N) equals 3. **Best viewed in color.**

estimation methods. Second, we study the region-level saliency modeling scheme and visualize the activation maps in different branches. Next, we show the effectiveness of Hyper Kernel Generator. Finally, we show the ablation studies on the proposed components in our method. Experiments are conducted with inputs at a resolution of 256 $\times$ 256, based on ResNet-50 backbone.

Prior Guidance Estimation. In this part, we explore several prior guidance G_{sal} to assist the learning process. As shown in Fig. 7, we show three kinds of saliency division results under different G_{sal} to illustrate the proposed method: including DT, Erode, Grad-Cam. Specifically, 'DT' [33] refers to calculating the nearest distance of each pixel to the boundary and decomposing the ground-truth map with given thresholds. 'Erode' means we iteratively erode the ground-truth label to get the annular saliency level division. 'Grad-Cam' [34] is a data-driven method that decompose the ground-truth salient objects according to the gradient response maps of each regions, which are provided by a standard classification model (i.e. ResNet-50 pre-trained on ImageNet).

As shown in Table 2, the method without extra guidance achieves 83% on $\mathcal{F}_\beta$ of DUTS-TE. Through explicit guidance on the sub-saliency masks with 'Erode'

Table 3: Performance (mean F-measure) and computational cost comparison of different approaches. These approaches use different strategies to generate kernel groups.

#	Settings	GFLOPs	Params	DUTS-TE	ECSSD
1	Static	13.2	25.1	.854	.927
2	Hyper Conv	15.3	31.4	.859	.927
3	Build-in Trans	15.7	61.0	.866	**.933**
4	**Hyper Trans**	15.1	32.4	**.867**	**.933**

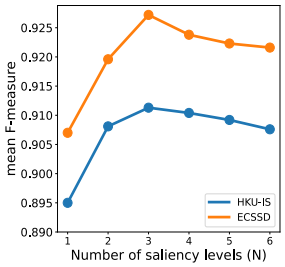

Fig. 8: Ablation on the number of saliency levels (from 1 to 6).

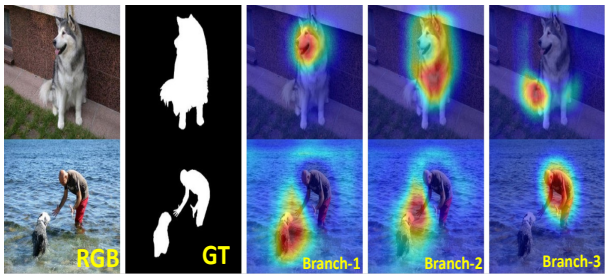

Fig. 9: The visualization of feature activation maps (Grad-Cam) from different branches in SHM_4. **Best viewed in color.**

and 'DT', the performance gains 1.9%, and 2.0%, respectively. Furthermore, when applying the 'Grad-Cam' to generate sub-saliency labels, the performance further increases to 85.4%. These results indicate that our framework supports various knowledge to mimic the hierarchy patterns for region-level modeling and the data-driven method (e.g., 'Grad-Cam') is more friendly to our framework.

Region-level Saliency Modeling. The number of saliency levels (N) is an important hyper-parameter. As shown in Fig. 8, we compare the mean F-measure on two datasets, and the qualitative results indicate that applying multi-branch learning patterns on the decomposed sub-saliency regions can significantly boost the performance. However, over decomposition may do harm to the global context, resulting in a slight performance drop. We achieve the best performance in our framework when N equals 3. Moreover, as shown in Fig. 9, we provide the visualization of feature activation maps (Grad-Cam) from different branches in SHM_4 (i.e., $\{H_4^{1\prime}, H_4^{2\prime}, H_4^{3\prime}\}$ in Eq. (3)). Different regions are activated in corresponding branches, which evidently justifies that our model achieves the region-level saliency hierarchy modeling by a divide-and-conquer strategy.

Sample-level Saliency Modeling. In order to verify the effectiveness of our proposed HKG, we conduct a series of experiments with different kernel generation strategies, as shown in Table 3. Our strategy is denoted as 'Hyper Trans', which uses shared hyper-kernels for all SHMs. 'Hyper Conv' means utilizing the convolutional architecture to generate the shared hyper-kernels. 'Build-in

Table 4: Ablation study of each module in our SHNet. 'Baseline' denotes the vanilla U-Net with ResNet-50 backbone. 'SHM (Static)' denotes the SHM with static kernels in the branches.

Settings	DUTS-TE		ECSSD		HKU-IS		PASCAL-S		DUT-O	
	$\mathcal{F}_\beta \uparrow$	$\mathcal{M} \downarrow$	$\mathcal{F}_\beta \uparrow$	$\mathcal{M} \downarrow$	$\mathcal{F}_\beta \uparrow$	$\mathcal{M} \downarrow$	$\mathcal{F}_\beta \uparrow$	$\mathcal{M} \downarrow$	$\mathcal{F}_\beta \uparrow$	$\mathcal{M} \downarrow$
Baseline (B)	.820	.041	.901	.041	.886	.036	.827	.067	.734	.060
B + SHM(Static)	.854	.034	.927	.031	.912	.028	.848	.059	.772	.050
B + SHM+HSG	**.867**	**.032**	**.933**	**.030**	**.918**	**.026**	**.849**	**.057**	**.784**	**.049**

Trans' stands for using multiple transformer architecture for generating different hyper-kernels for respective SHMs instead of the shared one. The results demonstrate that the proposed HKG is an effective way to produce the two-dimensional (layer, branch) kernel matrix for our decoder. Meanwhile, the shared hyper-kernels could reduce computational costs without any performance drop, which further verifies our module could excavate the hyper-knowledge for the diverse saliency patterns.

Effectiveness of the Proposals. As shown in Table 4, we use the vanilla U-Net with ResNet-50 backbone as our baseline model. 'SHM (Static)' indicates the Saliency Hierarchy Modules with static kernels in the multiple branches. The mean F-measure of 'SHM (Static)' is better than that of the baseline on DUTS-TE (85.4% against 82.0%) and DUT-O (77.2% against 73.4%). Moreover, with the HKG module, the performance is further improved to 78.4% in mean F-measure on DUT-O, and surpasses the state-of-the-art results with a low computational cost.

5 Conclusion

In this paper, we propose a framework named SHNet for SOD, which aims to model saliency hierarchically with generative kernels. We design a Saliency Hierarchy Module to model the hierarchical saliency levels in a given sample with the guide of prior knowledge. Furthermore, we design a Hyper Kernel Generator to automatically adjust our network parameters to the saliency divergence among different samples by generating cascaded kernel groups, which achieves a sample adaptive inference pattern. Extensive experiments demonstrate the effectiveness of our method on both performance and efficiency.

Acknowledgement

This work is supported in part by National Key Research and Development Program of China under Grant 2020AAA0107400, Zhejiang Provincial Natural Science Foundation of China under Grant LR19F020004, National Natural Science Foundation of China under Grant U20A20222.

References

1. Achanta, R., Hemami, S., Estrada, F., Susstrunk, S.: Frequency-tuned salient region detection. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 1597–1604. IEEE (2009) 3, 9
2. Achanta, R., Hemami, S.S., Estrada, F.J., Süsstrunk, S.: Frequency-tuned salient region detection. In: IEEE Conf. Comput. Vis. Pattern Recog. (2009) 3
3. Borji, A., Cheng, M.M., Jiang, H., Li, J.: Salient object detection: A benchmark. IEEE Trans. Image Process. **24**(12), 5706–5722 (2015) 9
4. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: Eur. Conf. Comput. Vis. pp. 213–229. Springer (2020) 3, 4
5. Chen, S., Tan, X., Wang, B., Hu, X.: Reverse attention for salient object detection. In: Eur. Conf. Comput. Vis. pp. 234–250 (2018) 3
6. Chen, Z., Xu, Q., Cong, R., Huang, Q.: Global context-aware progressive aggregation network for salient object detection. In: AAAI. vol. 34, pp. 10599–10606 (2020) 1
7. Cheng, B., Schwing, A., Kirillov, A.: Per-pixel classification is not all you need for semantic segmentation. Advances in Neural Information Processing Systems **34** (2021) 3
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 248–255. Ieee (2009) 9
9. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. Int. Conf. Learn. Represent. (2021) 4
10. Fan, D.P., Gong, C., Cao, Y., Ren, B., Cheng, M.M., Borji, A.: Enhanced-alignment measure for binary foreground map evaluation. IJCAI (2018) 9
11. Feng, M., Lu, H., Ding, E.: Attentive feedback network for boundary-aware salient object detection. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 1623–1632 (2019) 1, 10
12. Gu, Y.C., Gao, S.H., Cao, X.S., Du, P., Lu, S.P., Cheng, M.M.: inas: Integral nas for device-aware salient object detection. In: Int. Conf. Comput. Vis. pp. 4934–4944 (2021) 3, 4
13. Ha, D., Dai, A., Le, Q.V.: Hypernetworks. Int. Conf. Learn. Represent. (2016) 4
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 770–778 (2016) 5, 9
15. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. IEEE Trans. Pattern Anal. Mach. Intell. **20**(11), 1254–1259 (1998) 3
16. Ji, W., Li, X., Wei, L., Wu, F., Zhuang, Y.: Context-aware graph label propagation network for saliency detection. IEEE Trans. Image Process. **29**, 8177–8186 (2020) 3
17. Jia, X., De Brabandere, B., Tuytelaars, T., Gool, L.V.: Dynamic filter networks. Adv. Neural Inform. Process. Syst. **29**, 667–675 (2016) 4
18. Jiang, H., Wang, J., Yuan, Z., Wu, Y., Zheng, N., Li, S.: Salient object detection: A discriminative regional feature integration approach. In: Int. J. Comput. Vis. pp. 2083–2090 (2013) 3

19. Jiang, Z., Davis, L.S.: Submodular salient region detection. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 2043–2050 (2013). https://doi.org/10.1109/CVPR.2013.266 3
20. Li, G., Yu, Y.: Visual saliency based on multiscale deep features. In: CVPR. pp. 5455–5463. IEEE Computer Society (2015) 9
21. Li, J., Su, J., Xia, C., Ma, M., Tian, Y.: Salient object detection with purificatory mechanism and structural similarity loss. IEEE Trans. Image Process. **30**, 6855–6868 (2021) 3, 4
22. Li, Y., Hou, X., Koch, C., Rehg, J.M., Yuille, A.L.: The secrets of salient object segmentation. In: CVPR. pp. 280–287. IEEE Computer Society (2014) 9
23. Littwin, G., Wolf, L.: Deep meta functionals for shape representation. In: Int. Conf. Comput. Vis. pp. 1824–1833 (2019) 4
24. Liu, J.J., Hou, Q., Cheng, M.M.: Dynamic feature integration for simultaneous detection of salient object, edge, and skeleton. IEEE Trans. Image Process. **29**, 8652–8667 (2020) 3, 4
25. Liu, N., Zhang, N., Wan, K., Shao, L., Han, J.: Visual saliency transformer. In: Int. Conf. Comput. Vis. pp. 4722–4732 (2021) 3, 4, 10
26. Liu, T., Yuan, Z., Sun, J., Wang, J., Zheng, N., Tang, X., Shum, H.Y.: Learning to detect a salient object. IEEE Trans. Pattern Anal. Mach. Intell. **33**(2), 353–367 (2010) 3
27. Ma, M., Xia, C., Li, J.: Pyramidal feature shrinking for salient object detection. In: AAAI. vol. 35, pp. 2311–2318 (2021) 1, 3, 4
28. Mohammadi, S., Noori, M., Bahri, A., Majelan, S.G., Havaei, M.: Cagnet: Content-aware guidance for salient object detection. Pattern Recognition **103**, 107303 (2020) 10
29. Nirkin, Y., Wolf, L., Hassner, T.: Hyperseg: Patch-wise hypernetwork for real-time semantic segmentation. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 4061–4070 (2021) 4
30. Pang, Y., Zhao, X., Zhang, L., Lu, H.: Multi-scale interactive network for salient object detection. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 9413–9422 (2020) 3, 4, 10
31. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. Adv. Neural Inform. Process. Syst. (2019) 9
32. Qin, X., Zhang, Z., Huang, C., Gao, C., Dehghan, M., Jagersand, M.: Basnet: Boundary-aware salient object detection. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 7479–7489 (2019) 3
33. Rosenfeld, A., Pfaltz, J.L.: Distance functions on digital pictures. Pattern Recognition **1**(1), 33–61 (1968) 12
34. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Int. Conf. Comput. Vis. pp. 618–626 (2017) 8, 12
35. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. Int. Conf. Learn. Represent. (2014) 5, 9
36. Sun, P., Zhang, W., Wang, H., Li, S., Li, X.: Deep rgb-d saliency detection with depth-sensitive attention and automatic multi-modal fusion. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 1407–1417 (2021) 1, 3, 4
37. Tang, L., Li, B., Zhong, Y., Ding, S., Song, M.: Disentangled high quality salient object detection. In: Int. Conf. Comput. Vis. pp. 3580–3590 (2021) 3, 4

38. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Adv. Neural Inform. Process. Syst. pp. 5998–6008 (2017) 2, 4
39. Wang, B., Chen, Q., Zhou, M., Zhang, Z., Jin, X., Gai, K.: Progressive feature polishing network for salient object detection. In: AAAI. vol. 34, pp. 12128–12135 (2020) 3, 4
40. Wang, L., Lu, H., Wang, Y., Feng, M., Wang, D., Yin, B., Ruan, X.: Learning to detect salient objects with image-level supervision. In: CVPR. pp. 3796–3805. IEEE Computer Society (2017) 3, 9
41. Wang, T., Zhang, L., Lu, H., Sun, C., Qi, J.: Kernelized subspace ranking for saliency detection. In: Eur. Conf. Comput. Vis. pp. 450–466. Springer (2016) 3
42. Wei, J., Wang, S., Huang, Q.: F^3net: Fusion, feedback and focus for salient object detection. In: AAAI. pp. 12321–12328 (2020) 3, 4, 9, 10
43. Wei, J., Wang, S., Wu, Z., Su, C., Huang, Q., Tian, Q.: Label decoupling framework for salient object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13025–13034 (2020) 1, 3, 4, 10
44. Wu, R., Feng, M., Guan, W., Wang, D., Lu, H., Ding, E.: A mutual learning method for salient object detection with intertwined multi-supervision. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 10
45. Wu, Z., Su, L., Huang, Q.: Cascaded partial decoder for fast and accurate salient object detection. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 3907–3916 (2019) 1, 10
46. Wu, Z., Su, L., Huang, Q.: Stacked cross refinement network for edge-aware salient object detection. In: Int. Conf. Comput. Vis. pp. 7264–7273 (2019) 10
47. Wu, Z., Su, L., Huang, Q.: Decomposition and completion network for salient object detection **30**, 6226–6239 (2021) 3, 4, 10
48. Xu, B., Liang, H., Liang, R., Chen, P.: Locate globally, segment locally: A progressive architecture with knowledge review network for salient object detection. In: AAAI. vol. 35, pp. 3004–3012 (2021) 3, 4
49. Yan, Q., Xu, L., Shi, J., Jia, J.: Hierarchical saliency detection. In: CVPR. pp. 1155–1162. IEEE Computer Society (2013) 9
50. Yang, C., Zhang, L., Lu, H., Ruan, X., Yang, M.: Saliency detection via graph-based manifold ranking. In: CVPR. pp. 3166–3173. IEEE Computer Society (2013) 9
51. Zhang, L., Zhang, J., Lin, Z., Lu, H., He, Y.: Capsal: Leveraging captioning to boost semantics for salient object detection. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 6024–6033 (2019) 1
52. Zhang, M., Liu, T., Piao, Y., Yao, S., Lu, H.: Auto-msfnet: Search multi-scale fusion network for salient object detection. In: ACM Int. Conf. Multimedia (2021) 10
53. Zhao, J.X., Liu, J.J., Fan, D.P., Cao, Y., Yang, J., Cheng, M.M.: Egnet: Edge guidance network for salient object detection. In: Int. Conf. Comput. Vis. pp. 8779–8788 (2019) 1, 10
54. Zhao, T., Wu, X.: Pyramid feature attention network for saliency detection. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 3085–3094 (2019) 1
55. Zhao, X., Pang, Y., Zhang, L., Lu, H., Zhang, L.: Suppress and balance: A simple gated network for salient object detection. In: Eur. Conf. Comput. Vis. pp. 35–51. Springer (2020) 10

56. Zhao, Z., Xia, C., Xie, C., Li, J.: Complementary trilateral decoder for fast and accurate salient object detection. In: ACM Int. Conf. Multimedia. pp. 4967–4975 (2021) 3, 4, 10
57. Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P.H., et al.: Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 6881–6890 (2021) 4
58. Zhou, H., Xie, X., Lai, J.H., Chen, Z., Yang, L.: Interactive two-stream decoder for accurate and fast saliency detection. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 9141–9150 (2020) 10