Complementary Trilateral Decoder for Fast and Accurate Salient Object Detection

Zhirui Zhao[1], Changqun Xia[2,*], Chenxi Xie[1], Jia Li[1,2,*]
[1]State Key Laboratory of Virtual Reality Technology and Systems, SCSE, Beihang University, Beijing, China
[2]Peng Cheng Laboratory, Shenzhen, China

ABSTRACT

Salient object detection (SOD) has made great progress, but most of existing SOD methods focus more on performance than efficiency. Besides, the U-shape structure exists some drawbacks and there is still a lot of room for improvement. Therefore, we propose a novel framework to treat semantic context, spatial detail and boundary information separately in the decoder part. Specifically, we propose an efficient and effective Complementary Trilateral Decoder (CTD) for saliency detection with three branches: Semantic Path, Spatial Path and Boundary Path. These three branches are designed to solve the dilution of semantic information, loss of spatial information and absence of boundary information, respectively. These three branches are complementary to each other and we design three distinctive fusion modules to gradually merge them according to "coarse-fine-finer" strategy, which significantly improves the region accuracy and boundary quality. To facilitate the practical application in different environments, we provide two versions: CTDNet-18 (11.82M, 180FPS) and CTDNet-50 (24.63M, 110FPS). Experiments show that our model performs better than state-of-the-art approaches on five benchmarks, which achieves a favorable balance between speed and accuracy.

CCS CONCEPTS

• **Computing methodologies → Interest point and salient region detections**.

KEYWORDS

salient object detection, trilateral decoder, complementary, performance and efficiency

ACM Reference Format:
Zhirui Zhao[1], Changqun Xia[2,*], Chenxi Xie[1], Jia Li[1,2,*]. 2021. Complementary Trilateral Decoder for Fast and Accurate Salient Object Detection. In *Proceedings of the 29th ACM International Conference on Multimedia (MM '21), October 20–24, 2021, Virtual Event, China.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3474085.3475494

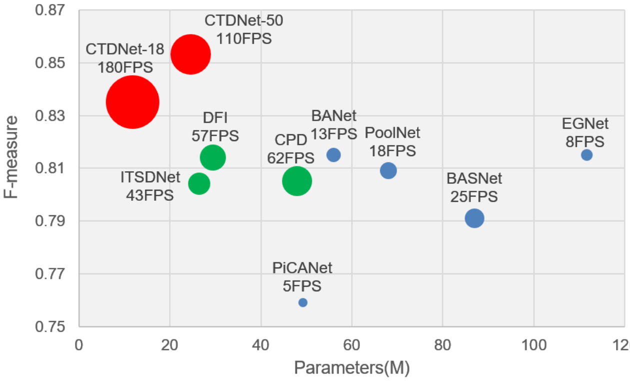


Figure 1: Comparisons of our proposed CTDNet with other ResNet-based SOD models in accuracy, parameters and speed. We calculate mF_β on DUTS-TE dataset as an example. The green circles represent real-time SOD methods, blue circles represent non-real-time SOD methods, and red circles represent our method. The size of circle indicates speed and larger circle indicates faster speed.

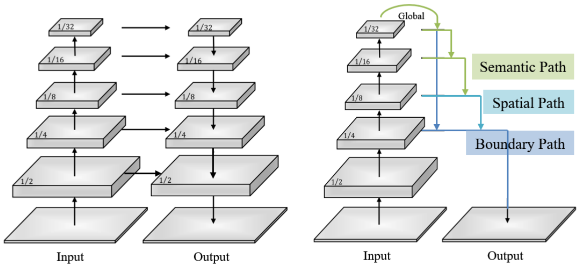
1 INTRODUCTION

The task of salient object detection (SOD) [1, 31] is to segment the most visually distinctive objects or regions in an image. As an efficient preprocessing technique, SOD is very important for many downstream computer vision tasks, like image retrieval [9], tracking [12], and segmentation [10].

Earlier traditional SOD algorithms [4, 13] mostly predicted saliency maps based on some hand-crafted features. Recently, the development of Convolutional Neural Networks (CNNs) [28], has greatly promoted the progress of SOD due to their powerful feature representation ability. However, most of existing SOD methods cannot achieve a favorable trade-off between efficiency and performance. On the one hand, some models tend to increase network depth and width to obtain better performance, causing heavy computational cost and slow inference speed. These methods often require a strong backbone (e.g., ResNet-50 or ResNet-101 [11]) and a complicated decoder, which makes them difficult to apply under resource constraints. As an example, EGNet [42] contains about 108M parameters and only runs at a speed of 9 FPS (see Fig. 1). On the other hand, some researches start to consider efficient saliency detection and try to compromise between speed and accuracy, such as CPD [35] and ITSDNet [45], but these models cannot obtain comparable performance (see Fig. 1). Therefore, it is significant and challenging to build a lightweight and fast SOD model with competitive performance.

[*] Correspondance should be addressed to Changqun Xia (E-mail: xiachq@pcl.ac.cn) and Jia Li (E-mail: jiali@buaa.edu.cn).
Website: https://cvteam.net/.

MM '21, October 20–24, 2021, Virtual Event, China
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8651-7/21/10…$15.00
https://doi.org/10.1145/3474085.3475494



(a) U-shape structure

(b) Our trilateral decoder

Figure 2: The U-shape structure gradually recovers the spatial information by leveraging lateral connections and top-down path in the decoder part, while our trilateral decoder separately treats semantic context, spatial detail and boundary information with three branches.

Among FCN-based SOD methods, the U-shape structure [27] has received the most attention and achieved good performance. The U-shape structure gradually recovers the high-resolution feature maps in the decoder part by leveraging a top-down path and lateral connections (see Fig. 2(a)), but it exists several drawbacks and there is still much room for improvement. First, a complete U-shape structure can increase the computational complexity and reduce the speed due to the large resolution of low-level features. Second, spatial information lost in the process of downsampling cannot be easily recovered only by merging the hierarchical features [39]. Third, semantic information of high-level features may be gradually diluted in the top-down path and global context information is also ignored, which probably produces incomplete segmentation results. Fourth, the U-shape structure lacks boundary information, which leads to poor boundary quality.

Based on the above observation, we abandon the traditional U-shape structure and propose to treat semantic context, spatial detail and boundary information separately in the decoder part to achieve a good balance between speed and accuracy. We propose an efficient and effective Complementary Trilateral Decoder (CTD) for saliency detection with three branches: Semantic Path, Spatial Path and Boundary Path (see Fig. 2(b)). These three branches are designed to solve the dilution of semantic information, loss of spatial information and absence of boundary information, respectively. These three parts are derived from different stages of the encoder and are complementary to each other, where the encoder is shared. We can gradually merge these three branches according to "coarse-fine-finer" strategy. Specifically, the Semantic Path is introduced to capture rich semantic context and global context with large receptive field, which can form an initial *coarse* saliency map with accurate locations of salient objects. In contrast, the Spatial Path is designed to preserve more spatial details. Both paths are combined to construct a comprehensive and powerful feature representation, which can produce a relative *fine* saliency map with precise structures of salient objects. As for the Boundary Path, we utilize low-level local features and high-level location features to extract salient boundary features with an extra edge supervision. Finally, we leverage the salient boundary features provided by the Boundary Path to further refine the fused features of the first two branches, which can generate a final *finer* saliency map with clear boundaries of salient objects.

Considering the characteristics and complementarity of these three branches, we propose three distinctive fusion modules to merge them effectively. We design a simple Feature Fusion Module (FFM) to fuse multi-level features efficiently for the Semantic Path and Boundary Path. Then we propose a novel Cross Aggregation Module (CAM) to merge the Semantic Path and Spatial Path. Besides, we design a Boundary Refinement Module (BRM) to further refine boundary. To facilitate the practical application in different environments, we provide two versions of our proposed method based on different backbone networks: CTDNet-18 and CTDNet-50. Experiments on five benchmarks demonstrate that CTDNet-18 achieves competitive or even better performance compared with large SOD models, and CTDNet-50 achieves the best performance. Moreover, our CTDNet-18 only has 11.82M parameters and runs at a speed of 180 FPS on a GTX 1080Ti GPU for 352×352 input images, which is smaller and faster than existing approaches with competitive performance. In general, our paper makes three major contributions:

- We propose a novel framework to treat semantic context, spatial detail and boundary information separately. To this end, we propose an efficient and effective Complementary Trilateral Decoder (CTD) for saliency detection with three branches: Semantic Path, Spatial Path and Boundary Path.
- We explore the characteristics and complementarity of these three branches. We further design three distinctive fusion modules to gradually merge these three paths according to "coarse-fine-finer" strategy, which significantly improves the region accuracy and boundary quality.
- We provide two versions: CTDNet-18 (11.82M, 180FPS) and CTDNet-50 (24.63M, 110FPS). Experiments show that our proposed method obtains highly competitive performance compared with 18 state-of-the-art methods, which achieves a good trade-off between efficiency and performance.

2 RELATED WORK

Recently, many FCN-based methods have achieved remarkable progress in the SOD task. As one of the most representative networks, U-Net [27] can generate accurate segmentation results by effectively combining low-level and high-level features, so many researches follow this U-shape structure for saliency detection. PiCANet [21] proposed a pixel-wise contextual attention network to learn informative context locations for each pixel. TDBU [32] learnt top-down and bottom-up saliency inference in a cooperative and iterative manner. ASNet [33] explored the relationship between fixation prediction and saliency detection by an efficient recurrent attention mechanism. MINet [24] focused on scale variation and class imbalance challenges by utilizing multi-level and multi-scale feature information. DASNet [41] proposed a depth-aware framework to improve the segmentation performance with depth constraints. PFSNet [22] proposed to aggregate adjacent feature nodes in pairs through layer by layer shrinkage, which can fuse details and semantics effectively, and discard interference information. However, these methods have brought huge amount of parameters

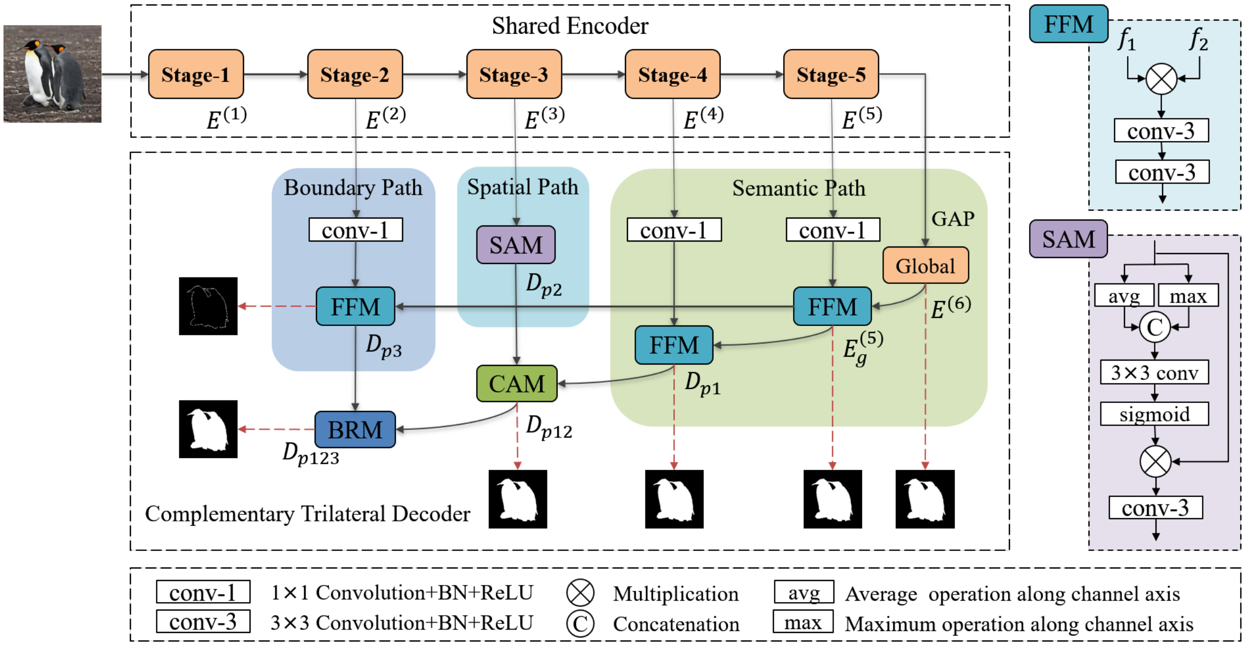


Figure 3: The framework of our proposed Complementary Trilateral Decoder (CTD) Network with three branches: Semantic Path, Spatial Path and Boundary Path, which treats semantic context, spatial detail and boundary information separately in the decoder part. The three parts share the same encoder and are derived from different stages of the encoder. The three branches are complementary to each other and we design three specific fusion modules to gradually merge them according to "coarse-fine-fine" strategy.

and high model complexity to achieve better performance, resulting in slow inference speed.

Recently, it has become more and more important to design a lightweight and fast model for saliency detection. PoolNet [20] fully exploited the pooling oprations based on the FPN [18] structure for real-time saliency detection. CPD [35] constructed a partial decoder for acceleration and utilized initial saliency map to refine features for better results. ITSDNet [45] proposed an interactive two-stream model to exploit contour and saliency information. Although smaller and faster than the previous large models, these methods cannot achieve comparable performance.

Because the U-shape structure may suffer from coarse object boundaries, some researches pay attention to boundary information by introducing an additional boundary-aware branch [44] or a boundary-aware loss function. C2SNet [16] borrowed contour knowledge for salient object detection. BASNet [26] proposed a boundary-aware model and a hybrid fusing loss for accurate salient object detection. EGNet [42] focused on the complementary information modeling of salient object and salient edge to improve the boundaries and localization. BANet [29] designed a boundary-aware model with successive dilation from the perspective of selectivity and invariance. PAGE [34] proposed pyramid attention structure for saliency detection and salient edge detection branch for boundary estimation. AFNet [8] proposed a boundary-aware model with multi-scale attentive feedback module and boundary-enhanced loss. SCRN [36] proposed an edge-aware network to bidirectionally pass messages between binary segmentation and edge map.

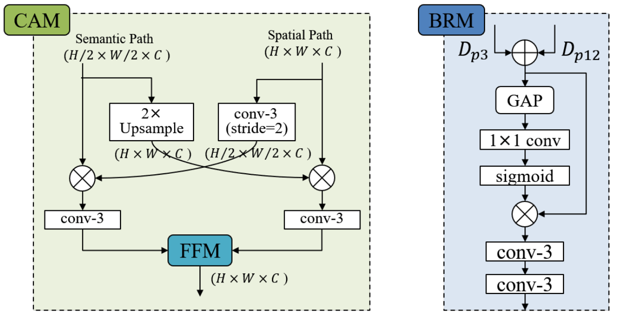
3 METHOD

Firstly, we outline the whole framework of our proposed CTDNet. Secondly, we introduce the effectiveness of these three branches in detail and how to merge them effectively. Finally, we describe the loss functions and supervision.

3.1 Overview of Network Architecture

Taking into account some drawbacks of the U-shape structure mentioned above, we propose a novel framework to treat semantic context, spatial detail and boundary information separately in the decoder. As Fig. 3 shows, we propose an efficient and effective Complementary Trilateral Decoder (CTD) for saliency detection with three branches: Semantic Path, Spatial Path and Boundary Path.

For the encoder, we adopt ResNet-50 [11] as the backbone network. In addition to ResNet-50, we also use shallow ResNet-18 [11] as the backbone network for lightweight and fast design concept. Both are pretrained on ImageNet [6] and encode multi-level features from different stages. The low-level features cost more computations due to the larger resolution, so we discard features of shallower layers for acceleration and only utilize the features of the last four stages that have strides of $\{4, 8, 16, 32\}$ with respect to the input image. For convenience, these four features can be expressed as $\left\{E^{(2)}, E^{(3)}, E^{(4)}, E^{(5)}\right\}$.



(a) Cross Aggregation Module (b) Boundary Refinement Module

Figure 4: The detailed structure of two fusion modules: CAM and BRM.

The decoder has three branches: Semantic Path, Spatial Path and Boundary Path. These three parts share the same encoder and are derived from different stages of the encoder. These three branches are designed to solve the dilution of semantic information, loss of spatial information and absence of boundary information, respectively. These three branches are complementary to each other and we design three distinctive fusion modules to gradually merge them according to "coarse-fine-fine" strategy, which significantly improves the region accuracy and boundary quality. More details are presented in Sec. 3.2 and 3.3.

3.2 Three Complementary Branches

3.2.1 Semantic Path. Both semantic context and global context are very important for saliency detection. However, the U-shape structure exists two issues: 1) the semantic information is gradually diluted in the top-down path; 2) the global context cannot be captured due to limited receptive field. Both these may lead to inaccurate locations of salient objects and incomplete segmentation results.

To solve these issues, we propose the Semantic Path to capture rich semantic context and global context with large receptive field, which can produce an initial coarse saliency map with accurate locations of salient objects. First, we embed a Global Average Pooling (GAP) layer on the tail of the backbone network, which can provide the maximum receptive field with the strongest global context. Then we apply a 1×1 convolution followed by a batch normalization and a ReLU activation function to $E^{(4)}$ and $E^{(5)}$, restricting the number of channels to 64 in order to reduce computation cost. Finally, we propose the Feature Fusion Module (FFM) to efficiently fuse the upsampled output of global pooling $E^{(6)}$ and the features of the last two stages, which forms a partial U-shape structure (see Fig. 3). The Semantic Path can be formulized as:

$$E_g^{(5)} = FFM_1(F_{1\times1}(E^{(5)}), Up(GAP(E^{(5)}))), \tag{1}$$

$$D_{p1} = FFM_2(F_{1\times1}(E^{(4)}), Up(E_g^{(5)})), \tag{2}$$

where $F_{1\times1}$ and Up represent 1×1 convolution and upsampling, respectively. The detailed structure of FFM is introduced in Sec. 3.3.

3.2.2 Spatial Path. While the Semantic Path captures rich semantic context and global context, the Spatial Path is devised to preserve more spatial details. The spatial information is also necessary for saliency detection, but it is seriously lost after multiple down-samplings and cannot be recovered perfectly by integrating the hierarchical features from the encoder. Therefore, we propose a Spatial Path to learn more discriminative feature representation from spatial dimension.

The Spatial Path is drawn from low-level features $E^{(3)}$ with large resolution (1/8 of the input size), which is beneficial to encode affluent spatial details. Specifically, we design the Spatial Attention Module (SAM) to refine features effectively (see Fig. 3). We first use average and maximum operations along the channel axis, generating two different single-channel spatial maps S_{avg} and S_{max}, respectively. Then we concatenate them and compute a spatial attention map by a 3×3 convolution and sigmoid function. The spatial attention map M_{sa} can re-weight the features $E^{(3)}$ from spatial dimension by element-wise multiplication. Finally, the refined features $E_{sa}^{(3)}$ are fed into a 3×3 convolution layer to squeeze the channels to 64. The Spatial Path can be formulized as:

$$S_{avg} = F_{avg}(E^{(3)}), S_{max} = F_{max}(E^{(3)}), \tag{3}$$

$$M_{sa} = \sigma(F_{3\times3}(Concat(S_{avg}, S_{max}))), \tag{4}$$

$$D_{p2} = F_{3\times3}(M_{sa} \otimes E^{(3)}) = F_{3\times3}(E_{sa}^{(3)}), \tag{5}$$

where F_{avg} and F_{max} denote average and maximum operations along the channel axis, respectively. $F_{3\times3}$ and $Concat$ represent 3×3 convolution and concatenation. σ and $\otimes$ represent the sigmoid function and element-wise multiplication.

3.2.3 Boundary Path. We observe that saliency maps produced by many existing SOD methods based on the U-shape structure have coarse boundaries. Therefore, we design a Boundary Path to improve boundary quality by utilizing boundary information explicitly. We can extract boundary features from low-level features $E^{(2)}$, which preserve better boundary information due to large resolution (1/4 of the input size). However, it is likely to bring noise and interference such as the boundaries of non-salient objects. Therefore, we exploit high-level location information as guidance to help enhance salient boundary features and suppress non-salient boundary features with an extra edge supervision (see Fig. 3).

Specifically, we first apply a 1×1 convolution followed by a batch normalization and a ReLU activation function to low-level features $E^{(2)}$. Then we upsample the high-level features $E_g^{(5)}$ (see Eq. (1)) to the same resolution as $E^{(2)}$ by bilinear interpolation. Finally, we use the FFM (see Sec. 3.3) to fuse them efficiently. In addition, we apply an explicit salient edge loss to supervise the Boundary Path explicitly. The Boundary Path can be formulized as:

$$D_{p3} = FFM_3(F_{1\times1}(E^{(2)}), Up(E_g^{(5)})). \tag{6}$$

3.3 Three Fusion Modules

3.3.1 Feature Fusion Module. We propose a simple fusion module FFM to fuse multi-level features efficiently for the Semantic Path and Boundary Path, as shown in Fig. 3. To be specific, FFM receives two inputs f_1 and f_2, and we adopt the multiplication operation to fuse these two features. Compared with addition and concatenation,

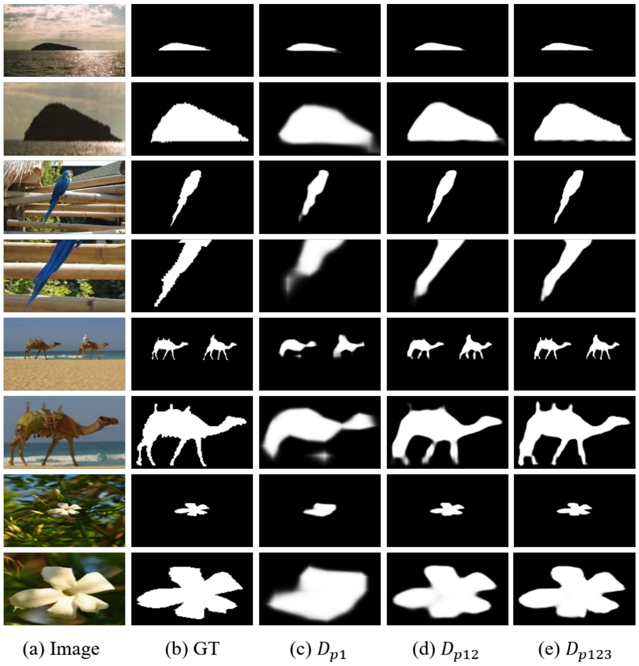


Figure 5: The saliency maps from different locations of our network. Each example contains two rows and the second row of each example denotes the zoom-in view of salient object. The results conform to "coarse-fine-finer" predictions along with the gradual combination of these three branches, which demonstrates the complementarity of these three branches.

the multiplication operation can avoid redundant information and suppress background noise. The fused features pass through two 3×3 convolution layers to obtain more robust feature representation. Note that each convolution is followed by a batch normalization and a ReLU activation function. The above process can be described as:

$$FFM(f_1, f_2) = F_{3\times3}(F_{3\times3}(f_1 \otimes f_2)). \tag{7}$$

3.3.2 Cross Aggregation Module. The output of the Semantic Path D_{p1} contains rich semantic information with global context, which can produce an initial coarse saliency map with accurate locations of salient objects (see Fig. 5(c)). In contrast, the output of the Spatial Path D_{p2} preserves more spatial details. Both paths are complementary to each other, so we design a novel fusion module CAM to merge these two branches effectively.

As Fig. 4(a) shows, the two inputs of CAM have different resolutions: $D_{p1} \in R^{\frac{H}{2}\times\frac{W}{2}\times C}$ and $D_{p2} \in R^{H\times W\times C}$. First, we perform the multi-scale transformation on each input. Specifically, we upsample D_{p1} to the same resolution as D_{p2} by bilinear interpolation and downsample D_{p2} to the same size as D_{p1} by a 3×3 convolution with stride 2, obtaining the corresponding features $D^{'}_{p1} \in R^{H\times W\times C}$ and $D^{'}_{p2} \in R^{\frac{H}{2}\times\frac{W}{2}\times C}$. Second, we perform cross aggregation on each scale by the multiplication operation and then apply a 3×3 convolution respectively to adapt them, which can capture multi-scale information and promote interaction between two branches. Note that each convolution is followed by a batch normalization and a ReLU activation function. Finally, the two fused features $C_1 \in R^{\frac{H}{2}\times\frac{W}{2}\times C}$ and $C_2 \in R^{H\times W\times C}$ are fed into the FFM to obtain the final output D_{p12}. By the proposed CAM, we construct a comprehensive and powerful feature representation, which can produce a relative fine saliency map with precise structures of salient objects (see Fig. 5(d)). The whole process can be described as:

$$D^{'}_{p1} = Up(D_{p1}), D^{'}_{p2} = Down_{3\times3}(D_{p2}), \tag{8}$$

$$C_1 = F_{3\times3}(D_{p1} \otimes D^{'}_{p2}), C_2 = F_{3\times3}(D_{p2} \otimes D^{'}_{p1}), \tag{9}$$

$$D_{p12} = FFM_4(Up(C_1), C_2), \tag{10}$$

where $Down_{3\times3}$ denotes downsampling operation using 3×3 convolution with stride 2.

3.3.3 Boundary Refinement Module. Although we obtain a relatively fine saliency map after merging the Semantic Path and Spatial Path, we can leverage the salient boundary information provided by the Boundary Path to further refine boundary. Therefore, we propose a fusion module BRM (see Fig. 4(b)) to merge D_{p12} and D_{p3}, which can generate a final finer saliency map with clear boundaries of salient objects (see Fig. 5(e)).

Specifically, we first combine the output features of the two branches by addition operation. Then we pool the fused features B_f to generate a feature vector and further calculate an attention vector to guide the feature learning by a 1×1 convolution and sigmoid function. The weight vector can reweight the fused features B_f for feature selection and refinement by multiplication operation. Finally, the refined features B_r are combined with B_f and then pass through two 3×3 convolution layers to further enhance feature representation. Note that each 3×3 convolution is followed by a batch normalization and a ReLU activation function. The above process can be described as:

$$B_f = Up(D_{p12}) + D_{p3}, \tag{11}$$

$$B_r = B_f \otimes \sigma(F_{1\times1}(GAP(B_f))), \tag{12}$$

$$D_{p123} = F_{3\times3}(F_{3\times3}(B_r + B_f)). \tag{13}$$

3.4 Loss Function

In SOD task, there are two common loss functions: BCE loss [5] and IoU loss [23]. BCE loss computes the error for each pixel between the prediction mask and the ground truth, which is formulated as:

$$\begin{split}\ell_{bce}(P,G) = -\sum_{i=1}^{H}\sum_{j=1}^{W}[G(i,j)log(P(i,j)) \\ +(1-G(i,j))log(1-P(i,j))],\end{split} \tag{14}$$

where $P(i,j)$ and $G(i,j)$ represent the pixel of prediction mask (P) and the ground truth (G) at location (i,j) in an image. W and H are the width and height of the image, respectively. IoU loss is used to measure the similarity of structure instead of focusing on single pixel. We adopt the following form:

$$\ell_{iou}(P,G) = 1 - \frac{\sum_{i=1}^{H}\sum_{j=1}^{W} G(i,j)P(i,j)}{\sum_{i=1}^{H}\sum_{j=1}^{W}[G(i,j) + P(i,j) - G(i,j)P(i,j)]}. \tag{15}$$

Table 1: Quantitative comparisons with state-of-the-art SOD models on five benchmarks in terms of parameters, speed, mF_β, MAE and E_m. The best two results are shown in red and green, respectively.

Method	Params (M)	Speed (FPS)	ECSSD			PASCAL-S			DUTS-TE			HKU-IS			DUT-OMRON		
			mF_β	MAE	E_m	mF_β	MAE	E_m	mF_β	MAE	E_m	mF_β	MAE	E_m	mF_β	MAE	E_m
VGG-based models																	
C2SNet$_{18}$	158.86	30	.864	.055	.914	.758	.080	.839	.716	.063	.846	.851	.048	.927	.683	.072	.829
RAS$_{18}$	21.23	35	.889	.056	.914	.777	.101	.829	.751	.059	.861	.871	.045	.929	.713	.062	.846
PiCANet$_{18}$	38.32	7	.885	.046	.910	.789	.077	.828	.749	.054	.852	.870	.042	.934	.710	.068	.834
BMPM$_{18}$	75.07	22	.868	.045	.914	.758	.073	.836	.745	.049	.860	.871	.039	.937	.692	.064	.837
EGNet$_{19}$	108.07	9	.913	.041	.923	.809	.076	.843	.800	.044	.880	.893	.035	.945	.744	.057	.858
PoolNet$_{19}$	52.51	32	.910	.042	.921	.806	.071	.839	.799	.042	.881	.894	.033	.948	.739	.056	.858
PAGE$_{19}$	47.40	25	.906	.042	.920	.806	.075	.841	.777	.052	.869	.882	.037	.940	.736	.062	.853
AFNet$_{19}$	35.99	26	.908	.042	.918	.820	.070	.850	.793	.046	.879	.888	.036	.942	.738	.057	.853
CPD$_{19}$	29.10	66	.915	.040	.922	.820	.072	.850	.813	.043	.892	.896	.033	.945	.745	.057	.863
GateNet$_{20}$	-	-	.896	.041	.921	.797	.067	.849	.783	.045	.881	.889	.036	.945	.723	.061	.848
ITSDNet$_{20}$	17.08	48	.875	.040	.917	.773	.067	.844	.798	.042	.892	.890	.035	.944	.745	.063	.855
ResNet-based models																	
PiCANet$_{18}$	49.31	5	.886	.046	.913	.792	.074	.832	.759	.051	.862	.870	.043	.936	.717	.065	.841
BANet$_{19}$	56.02	13	.923	.035	.928	.823	.069	.852	.815	.040	.892	.900	.032	.950	.746	.059	.860
EGNet$_{19}$	111.78	8	.920	.037	.927	.817	.073	.848	.815	.039	.891	.901	.031	.950	.755	.053	.867
SCRN$_{19}$	25.32	32	.918	.038	.926	.826	.064	.857	.809	.040	.888	.896	.034	.949	.746	.056	.863
PoolNet$_{19}$	68.16	18	.915	.039	.924	.815	.074	.848	.809	.040	.889	.899	.032	.949	.747	.056	.863
CPD$_{19}$	47.97	62	.917	.037	.925	.820	.070	.849	.805	.043	.887	.891	.034	.944	.747	.056	.866
BASNet$_{19}$	87.03	25	.880	.037	.921	.771	.075	.846	.791	.048	.884	.895	.032	.946	.756	.056	.869
GateNet$_{20}$	-	-	.916	.040	.924	.819	.067	.851	.807	.040	.889	.899	.033	.949	.746	.055	.862
U2Net$_{20}$	46.21	30	.892	.033	.924	.770	.073	.842	.792	.045	.886	.896	.031	.948	.761	.054	.871
DFI$_{20}$	29.57	57	.920	.038	.924	.830	.064	.855	.814	.039	.892	.901	.031	.951	.752	.055	.865
GCPANet$_{20}$	67.05	50	.919	.035	.920	.827	.061	.847	.817	.038	.891	.898	.031	.949	.748	.056	.860
ITSDNet$_{20}$	26.55	43	.895	.035	.927	.785	.071	.850	.804	.041	.895	.899	.031	.952	.756	.061	.863
MINet$_{20}$	162.38	31	.924	.033	.927	.829	.063	.851	.828	.037	.898	.909	.029	.953	.755	.055	.865
Our CTDNet																	
CTDNet-18	11.82	180	.920	.037	.921	.831	.065	.857	.835	.037	.902	.916	.028	.955	.767	.052	.873
CTDNet-50	24.63	110	.927	.032	.925	.841	.061	.861	.853	.034	.909	.919	.027	.955	.779	.052	.875

As described above, our model is deeply supervised with six outputs. All outputs pass through a 3×3 convolution and sigmoid function to convert the feature maps to the corresponding single-channel prediction masks. For $D_{p123}, D_{p12}, D_{p1}, E_g^{(5)}$ and $E^{(6)}$, we use BCE loss and IoU loss together to supervise these five prediction masks (see Eq. (16)), while for D_{p3}, we only use BCE loss to supervise the boundary prediction mask (P_b). Note that the ground truth of salient boundary (G_b) can be easily obtained from the ground truth of salient objects.

$$\ell(P, G) = \ell_{iou}(P, G) + \beta \ell_{bce}(P, G), \tag{16}$$

where β is a hyperparameter to balance the weight between the two loss functions, so that the network can achieve better performance. In our paper, the parameter β is set to 0.6. The total loss function is denoted:

$$L(P, P_b, G, G_b) = \ell_{bce}(P_b, G_b) + \sum_{k=1}^{5} \alpha_k \ell(P^k, G), \tag{17}$$

where α_k denotes the weight of the $k - th$ loss term.

4 EXPERIMENTS

Firstly, we describe five popular SOD datasets and evaluation metrics. Secondly, we introduce the implementation details. Finally, we present extensive experimental results to demonstrate the superiority and efficiency of our proposed model.

4.1 Datasets

We conduct experiments on five standard benchmark datasets: ECSSD (1,000) [37], PASCAL-S (850) [17], DUTS (15,552) [30], HKU-IS (4,447) [15] and DUT-OMRON (5,168) [38], all of which contain complex scenarios with one or more salient objects. In particular, DUTS includes 10,553 images for training (DUTS-TR) and 5,019 images for testing (DUTS-TE). We train our method on DUTS-TR and test our method on other datasets.

4.2 Evaluation Metrics

We adopt three evaluation metrics: Mean Absolute Error (MAE), F-measure and E-measure [7] to quantitatively evaluate the performance. MAE represents the pixel-wise average absolute difference

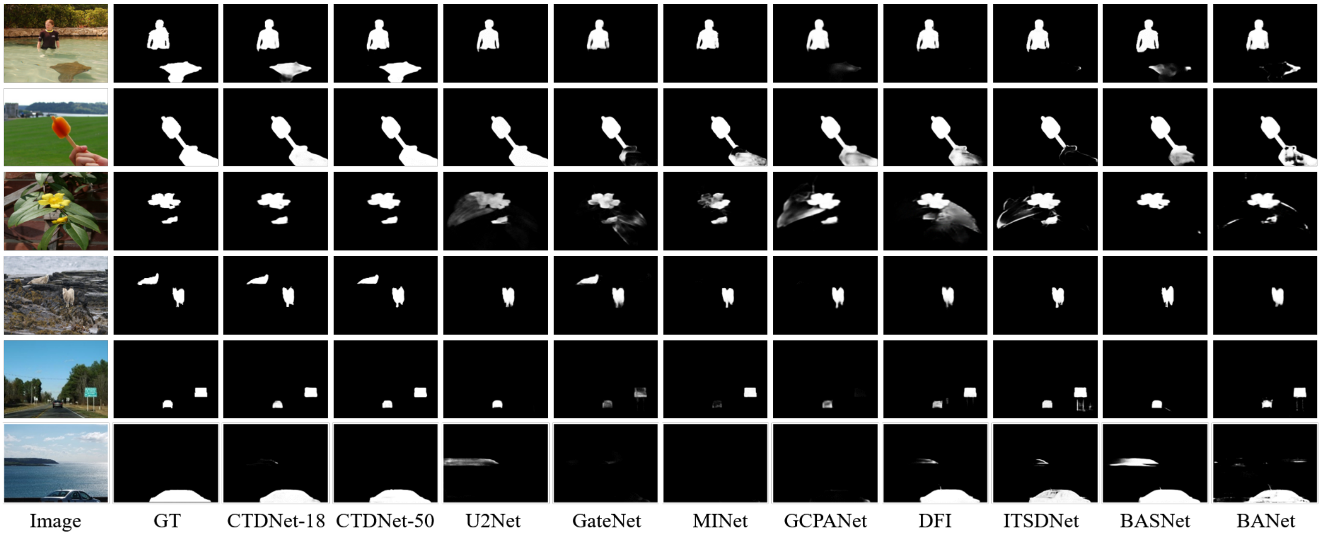


Figure 6: Qualitative comparison of our model with existing state-of-the-art SOD models in some challenging scenarios.

between prediction mask and ground truth:

$$MAE = \frac{1}{W \times H} \sum_{i=1}^{H} \sum_{j=1}^{W} |P(i,j) - G(i,j)|, \quad (18)$$

where P and G represent the prediction mask and the corresponding ground truth, respectively. W, H are the width and height of the image. Smaller MAE indicates better performance. F-measure (F_β) takes both precision and recall into account:

$$F_\beta = \frac{(1+\beta^2) \cdot Precision \cdot Recall}{\beta^2 \cdot Precision + Recall}, \quad (19)$$

where β^2 is set to 0.3 to emphasize the precision over recall. Larger F_β indicates better performance. We choose the mean F-measure (mF_β) in our paper. E-measure (E_m) measures the structural similarity between the prediction mask and the ground truth.

4.3 Implementation Details

We implement our method by PyTorch and conduct experiments on a NVIDIA 1080Ti GPU. We adopt ResNet-18 and ResNet-50 [11] pre-trained on ImageNet [6] as backbone networks, respectively. In the training period, all training images are resized to 352×352 with random cropping and random horizontal flipping to feed into the proposed model. We use stochastic gradient descent (SGD) optimizer with momentum of 0.9 and weight decay of 5e-4 to train our model. The batch size is 32 and training epoch is 40. We use the warm-up and linear decay learning rate strategy with the maximum learning rate 5e-3 for pre-trained backbone and 5e-2 for the rest of network. During the inference period, each image is simply resized to 352×352 to predict saliency map without any post-processing (e.g., CRF [14]).

4.4 Comparison results

To prove the effectiveness of our method, we compare with 18 state-of-the-art SOD models, including C2SNet [16], RAS [2], PiCANet [21], BMPM [40], BANet [29], EGNet [42], SCRN [36], PoolNet [20], PAGE [34], AFNet [8], CPD [35], BASNet [26], GateNet [43], DFI [19], ITSDNet [45], GCPANet [3], MINet [24] and U2Net [25]. For a fair comparison, we use saliency maps released by the authors and evaluate them with the same Matlab code.

4.4.1 Qualitative Comparison. To intuitively show the advantages of our model, we provide some visual examples of various SOD models, as shown in Fig. 6. We can observe that our method CTDNet-18 and CTDNet-50 can generate more complete and more accurate segmentation results than other counterparts. It can handle various challenging scenarios, such as multiple salient objects (row 1 ,4 and 5), fine structure (row 2 and 3), cluttered backgrounds (row 3), small objects (row 4 and 5) and foreground interference (row 6). In addition, we do not use any post-processing to obtain these results. Therefore, our model shows its effectiveness and robustness in processing complicated images.

4.4.2 Quantitative Comparison. Tab. 1 shows the quantitative results on five popular datasets in terms of mF_β, MAE and E_m. In addition, we also list the parameters and speed of each method to measure efficiency. To facilitate the practical application in different environments, we adopt ResNet-18 and ResNet-50 as backbones respectively and name our model CTDNet-18 and CTDNet-50 accordingly. On the one hand, our approach CTDNet-18 outperforms all the VGG-based SOD models and achieves comparable or even better performance than ResNet-based SOD models. More importantly, CTDNet-18 only has 11.82M parameters and runs at a speed of 180 FPS on one GTX 1080Ti GPU for 352×352 input images, which surpasses the existing approaches by a large margin. On the other hand, our approach CTDNet-50 obtains the best performance against state-of-the-art methods under almost all evaluation metrics on five benchmarks. Moreover, CTDNet-50 runs at a 110 FPS speed with 24.63M parameters, which is much smaller and faster than the existing ResNet-based SOD methods. In conclusion, our

Table 2: The ablation study of our proposed components. The backbone network takes ResNet-18 as an example. By adding each module gradually, our model achieves the best performance.

Base	FFM	Global	SAM	CAM	BRM	Params (M)	Speed (FPS)	DUTS-TE			DUT-OMRON		
								mF_β	MAE	E_m	mF_β	MAE	E_m
✓						11.39	240	.788	.048	.874	.718	.066	.840
✓	✓					11.46	230	.799	.045	.880	.730	.063	.847
✓	✓	✓				11.57	210	.813	.041	.892	.749	.057	.860
✓	✓	✓	✓			11.63	200	.821	.040	.895	.756	.055	.867
✓	✓	✓	✓	✓		11.78	188	.828	.039	.896	.761	.054	.870
✓	✓	✓	✓	✓	✓	11.82	180	.835	.037	.902	.767	.052	.873

Table 3: The complementarity of these three branches. D_{p1}, D_{p12} and D_{p123} denote the Semantic Path, the combination of both Semantic Path and Spatial Path, the combination of these three branches, respectively.

Merge	DUTS-TE			DUT-OMRON		
	mF_β	MAE	E_m	mF_β	MAE	E_m
D_{p1}	.762	.050	.871	.711	.063	.848
D_{p12}	.820	.039	.896	.756	.054	.869
D_{p123}	.835	.037	.902	.767	.052	.873

model achieves a favorable trade-off between speed and accuracy, which clearly demonstrates its superiority and efficiency.

4.5 Ablation Study

Firstly, we investigate the complementarity of these three branches. Secondly, we verify the effectiveness of our proposed components. All experiments are conducted on DUTS-TE and DUT-OMRON datasets based on ResNet-18 network.

4.5.1 The complementarity of these three branches. To demonstrate the complementarity and necessity of these three branches, we conduct experiments both qualitatively and quantitatively. As shown in Tab. 3, when merging the Semantic Path D_{p1} and Spatial Path D_{p2}, the performance can be greatly improved. Moreover, the performance can be further boosted by merging D_{p12} and D_{p3}, which benefits from the salient boundary information provided by the Boundary Path D_{p3}. In addition, we visualize some examples in Fig. 5. Each example contains two rows and the second row of each example denotes the zoom-in view of salient object. As we can see, the produced saliency maps conform to "coarse-fine-finer" predictions along with the gradual combination of these three branches. Column 3 represents initial coarse saliency maps produced by D_{p1} with accurate locations of salient objects. Column 4 represents relatively fine saliency maps produced by D_{p12} with precise structures of salient objects. Column 5 represents final finer saliency maps produced by D_{p123} with clear boundaries of salient objects. Obviously, experimental results verify the complementarity and necessity of these three branches.

4.5.2 The effectiveness of our proposed components. To demonstrate the effectiveness of our proposed components, we conduct ablation experiments by gradually adding them. First, we replace all the proposed fusion modules with simple addition operation followed by the 3×3 convolution to construct a baseline network, which still maintains three branches in the decoder. Second, we gradually add the FFM and global context for the Semantic Path and Boundary Path. Then we add the SAM in the Spatial Path. Finally, we use the proposed fusion modules CAM and BRM to merge these three branches. As shown in Tab. 2, our method achieves the best performance when all modules are contained, which demonstrates the effectiveness and necessity of each module.

5 CONCLUSION

In this paper, we first reveal that the existing SOD methods cannot achieve a good balance between speed and accuracy. Then we analyze the drawbacks of the U-shape structure. To this end, we propose to treat semantic context, spatial detail and boundary information separately in the decoder. Based on this idea, we propose an efficient and effective Complementary Trilateral Decoder for saliency detection with three branches: Semantic Path, Spatial Path and Boundary Path. These three branches are designed to solve the dilution of semantic information, loss of spatial information and absence of boundary information, respectively. These three branches are complementary to each other and we design three distinctive fusion modules to gradually merge them according to "coarse-fine-finer" strategy, which significantly improves the region accuracy and boundary quality. To facilitate the practical application in different environments, we provide two versions: CTDNet-18 (11.82M, 180FPS) and CTDNet-50 (24.63M, 110FPS). Experiments demonstrate that our proposed method performs better than state-of-the-art methods on five benchmarks, which achieves a favorable trade-off between efficiency and performance.

ACKNOWLEDGMENTS

This work was supported by grants from National Natural Science Foundation of China (No. 61922006) and CAAI-Huawei MindSpore Open Fund.

REFERENCES

[1] Ali Borji, Ming-Ming Cheng, Qibin Hou, Huaizu Jiang, and Jia Li. 2019. Salient object detection: A survey. *Computational visual media* (2019), 1–34.

[2] Shuhan Chen, Xiuli Tan, Ben Wang, and Xuelong Hu. 2018. Reverse attention for salient object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 234–250.

[3] Zuyao Chen, Qianqian Xu, Runmin Cong, and Qingming Huang. 2020. Global context-aware progressive aggregation network for salient object detection. *arXiv preprint arXiv:2003.00651* (2020).

[4] Ming-Ming Cheng, Niloy J Mitra, Xiaolei Huang, Philip HS Torr, and Shi-Min Hu. 2014. Global contrast based salient region detection. *IEEE transactions on pattern analysis and machine intelligence* 37, 3 (2014), 569–582.

[5] Pieter-Tjerk De Boer, Dirk P Kroese, Shie Mannor, and Reuven Y Rubinstein. 2005. A tutorial on the cross-entropy method. *Annals of operations research* 134, 1 (2005), 19–67.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.

[7] Deng-Ping Fan, Cheng Gong, Yang Cao, Bo Ren, Ming-Ming Cheng, and Ali Borji. 2018. Enhanced-alignment measure for binary foreground map evaluation. *arXiv preprint arXiv:1805.10421* (2018).

[8] Mengyang Feng, Huchuan Lu, and Errui Ding. 2019. Attentive feedback network for boundary-aware salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1623–1632.

[9] Yuan Gao, Miaojing Shi, Dacheng Tao, and Chao Xu. 2015. Database saliency for fast image retrieval. *IEEE Transactions on Multimedia* 17, 3 (2015), 359–369.

[10] Robert M Haralick and Linda G Shapiro. 1985. Image segmentation techniques. *Computer vision, graphics, and image processing* 29, 1 (1985), 100–132.

[11] K He, X Zhang, S Ren, and J Sun. 2016. Deep residual learning for image recognition. 2016 IEEE Conf Comput VisPattern Recognit. 2016: 770-778 https://doi.org/10.1109. CVPR.

[12] Seunghoon Hong, Tackgeun You, Suha Kwak, and Bohyung Han. 2015. Online tracking by learning discriminative saliency map with convolutional neural network. In *International conference on machine learning*. 597–606.

[13] Huaizu Jiang, Jingdong Wang, Zejian Yuan, Yang Wu, Nanning Zheng, and Shipeng Li. 2013. Salient object detection: A discriminative regional feature integration approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2083–2090.

[14] Philipp Krähenbühl and Vladlen Koltun. 2011. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*. 109–117.

[15] Guanbin Li and Yizhou Yu. 2015. Visual saliency based on multiscale deep features. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5455–5463.

[16] Xin Li, Fan Yang, Hong Cheng, Wei Liu, and Dinggang Shen. 2018. Contour knowledge transfer for salient object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 355–370.

[17] Yin Li, Xiaodi Hou, Christof Koch, James M Rehg, and Alan L Yuille. 2014. The secrets of salient object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 280–287.

[18] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. 2017. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2117–2125.

[19] Jiang-Jiang Liu, Qibin Hou, and Ming-Ming Cheng. 2020. Dynamic Feature Integration for Simultaneous Detection of Salient Object, Edge and Skeleton. *arXiv preprint arXiv:2004.08595* (2020).

[20] Jiang-Jiang Liu, Qibin Hou, Ming-Ming Cheng, Jiashi Feng, and Jianmin Jiang. 2019. A simple pooling-based design for real-time salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3917–3926.

[21] Nian Liu, Junwei Han, and Ming-Hsuan Yang. 2018. Picanet: Learning pixel-wise contextual attention for saliency detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3089–3098.

[22] Mingcan Ma, Changqun Xia, and Jia Li. 2021. Pyramidal Feature Shrinking for Salient Object Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 2311–2318.

[23] Gellért Máttyus, Wenjie Luo, and Raquel Urtasun. 2017. Deeproadmapper: Extracting road topology from aerial images. In *Proceedings of the IEEE International Conference on Computer Vision*. 3438–3446.

[24] Youwei Pang, Xiaoqi Zhao, Lihe Zhang, and Huchuan Lu. 2020. Multi-Scale Interactive Network for Salient Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9413–9422.

[25] Xuebin Qin, Zichen Zhang, Chenyang Huang, Masood Dehghan, Osmar R Zaiane, and Martin Jagersand. 2020. U2-Net: Going deeper with nested U-structure for salient object detection. *Pattern Recognition* 106 (2020), 107404.

[26] Xuebin Qin, Zichen Zhang, Chenyang Huang, Chao Gao, Masood Dehghan, and Martin Jagersand. 2019. Basnet: Boundary-aware salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7479–7489.

[27] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*. Springer, 234–241.

[28] Evan Shelhamer, Jonathan Long, and Trevor Darrell. 2017. Fully convolutional networks for semantic segmentation. *IEEE transactions on pattern analysis and machine intelligence* 39, 4 (2017), 640–651.

[29] Jinming Su, Jia Li, Yu Zhang, Changqun Xia, and Yonghong. Tian. 2019. Selectivity or Invariance: Boundary-aware Salient Object Detection. In *ICCV*.

[30] Lijun Wang, Huchuan Lu, Yifan Wang, Mengyang Feng, Dong Wang, Baocai Yin, and Xiang Ruan. 2017. Learning to detect salient objects with image-level supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 136–145.

[31] Wenguan Wang, Qiuxia Lai, Huazhu Fu, Jianbing Shen, Haibin Ling, and Ruigang Yang. 2021. Salient object detection in the deep learning era: An in-depth survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[32] Wenguan Wang, Jianbing Shen, Ming-Ming Cheng, and Ling Shao. 2019. An iterative and cooperative top-down and bottom-up inference network for salient object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5968–5977.

[33] Wenguan Wang, Jianbing Shen, Xingping Dong, Ali Borji, and Ruigang Yang. 2019. Inferring salient objects from human fixations. *IEEE transactions on pattern analysis and machine intelligence* 42, 8 (2019), 1913–1927.

[34] Wenguan Wang, Shuyang Zhao, Jianbing Shen, Steven CH Hoi, and Ali Borji. 2019. Salient object detection with pyramid attention and salient edges. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1448–1457.

[35] Zhe Wu, Li Su, and Qingming Huang. 2019. Cascaded partial decoder for fast and accurate salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3907–3916.

[36] Zhe Wu, Li Su, and Qingming Huang. 2019. Stacked cross refinement network for edge-aware salient object detection. In *Proceedings of the IEEE International Conference on Computer Vision*. 7264–7273.

[37] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. 2013. Hierarchical saliency detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1155–1162.

[38] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. 2013. Saliency detection via graph-based manifold ranking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3166–3173.

[39] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. 2018. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proceedings of the European conference on computer vision (ECCV)*. 325–341.

[40] Lu Zhang, Ju Dai, Huchuan Lu, You He, and Gang Wang. 2018. A bi-directional message passing model for salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1741–1750.

[41] Jiawei Zhao, Yifan Zhao, Jia Li, and Xiaowu Chen. 2020. Is depth really necessary for salient object detection?. In *Proceedings of the 28th ACM International Conference on Multimedia*. 1745–1754.

[42] Jia-Xing Zhao, Jiang-Jiang Liu, Deng-Ping Fan, Yang Cao, Jufeng Yang, and Ming-Ming Cheng. 2019. EGNet: Edge guidance network for salient object detection. In *Proceedings of the IEEE International Conference on Computer Vision*. 8779–8788.

[43] Xiaoqi Zhao, Youwei Pang, Lihe Zhang, Huchuan Lu, and Lei Zhang. 2020. Suppress and balance: A simple gated network for salient object detection. *arXiv preprint arXiv:2007.08074* (2020).

[44] Yifan Zhao, Jia Li, Yu Zhang, and Yonghong Tian. 2019. Multi-class part parsing with joint boundary-semantic awareness. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9177–9186.

[45] Huajun Zhou, Xiaohua Xie, Jian-Huang Lai, Zixuan Chen, and Lingxiao Yang. 2020. Interactive Two-Stream Decoder for Accurate and Fast Saliency Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9141–9150.