



# Hierarchical Dynamic Filtering Network for RGB-D Salient Object Detection

Youwei Pang<sup>1</sup>, Lihe Zhang<sup>1(✉)</sup>, Xiaoqi Zhao<sup>1</sup>, and Huchuan Lu<sup>1,2</sup>

<sup>1</sup> Dalian University of Technology, Dalian, China  
{lartpang, zxq}@mail.dlut.edu.cn

<sup>2</sup> Peng Cheng Laboratory, Shenzhen, China  
{zhanglihe, lhchuan}@dlut.edu.cn

**Abstract.** The main purpose of RGB-D salient object detection (SOD) is how to better integrate and utilize cross-modal fusion information. In this paper, we explore these issues from a new perspective. We integrate the features of different modalities through densely connected structures and use their mixed features to generate dynamic filters with receptive fields of different sizes. In the end, we implement a kind of more flexible and efficient multi-scale cross-modal feature processing, i.e. dynamic dilated pyramid module. In order to make the predictions have sharper edges and consistent saliency regions, we design a hybrid enhanced loss function to further optimize the results. This loss function is also validated to be effective in the single-modal RGB SOD task. In terms of six metrics, the proposed method outperforms the existing twelve methods on eight challenging benchmark datasets. A large number of experiments verify the effectiveness of the proposed module and loss function. Our code, model and results are available at <https://github.com/lartpang/HDFNet>.

**Keywords:** RGB-D Salient Object Detection · Cross-modal fusion · Dynamic dilated pyramid module · Hybrid enhanced loss

## 1 Introduction

Salient object detection (SOD) aims to model the mechanism of human visual attention and mine the most salient objects or regions in data such as images or videos. SOD has been widely applied in many computer vision tasks, such as scene classification [38], video segmentation [13], semantic segmentation [44], foreground map evaluation [10, 11] visual tracking [30], person re-identification [39], light field image segmentation [43] and so on.

With the advent of the fully convolutional network [29], deep learning-based SOD models [17, 27] have made great progress. Some methods [22, 33, 50, 51]

---

**Electronic supplementary material** The online version of this chapter ([https://doi.org/10.1007/978-3-030-58595-2\\_15](https://doi.org/10.1007/978-3-030-58595-2_15)) contains supplementary material, which is available to authorized users.

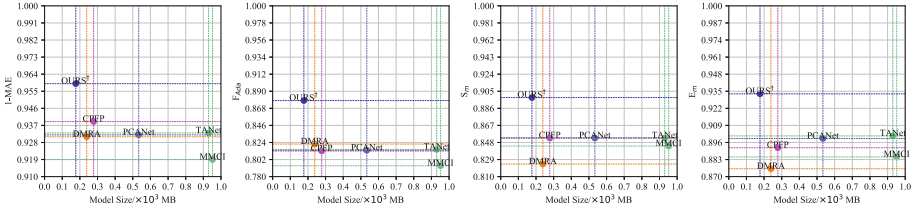


Fig. 1. Comparisons in model size and accuracy.

have achieved very good performance on the existing benchmark datasets. However, these works are mainly based on RGB data. They still face severe challenges when handling the cluttered or low-contrast scenes. Recently, some works [2, 4, 6, 8, 14, 37, 53] introduce the depth data as an aid to further improve the detection performance. The depth information can more intuitively express spatial structures of the objects in a scene and provide a powerful supplement for the detection and recognition of salient objects. Using complementary modal cues, the scene can be further deeply and intelligently understood. However, limited by the way of using the depth information, RGB-D salient object detection is still great challenging.

It is well known that RGB images contain rich appearance and detail information while depth images contain more spatial structure information. They complement each other for many vision tasks. RGB-D SOD approaches aim to formulate cross-modal fusion in different manners. Most of them integrate depth and RGB features by element-wise addition [4, 36], concatenation [3, 12] and convolution operations [2, 42]. Some methods compute attention map [49] or saliency map [42] via a shallow or deep CNN network from pure depth images. Because of using the fixed parameters for different samples during the testing phase, the generalization capability of these models is weakened. Moreover, for dense prediction task, the loss in each spatial position is usually different. Thus, the actual optimization direction of gradients in different positions may be varying. The weight-sharing convolution operation across different positions, which is used in the existing methods, causes that the training process of each parameter relies on the global gradient. This forces the network to learn trade-off and sub-optimal parameters. To address these problems, we propose a dynamic dilated pyramid module (DDPM), which uses RGB-depth mixed features to adaptively adjust convolution kernels for different input samples and processing locations. These kernels can capture rich semantic cues at multiple scales with the help of the pyramid structure and the dilated convolution. This design is capable of making more efficient convolution operations for current RGB features and promotes the network to obtain more flexible and targeted features for saliency prediction.

Early deep learning-based SOD models [14], which use fully connected layers, destroy the spatial structure of the data. This issue is alleviated to some extent by using the fully convolutional network. But the intrinsic gridding operation and

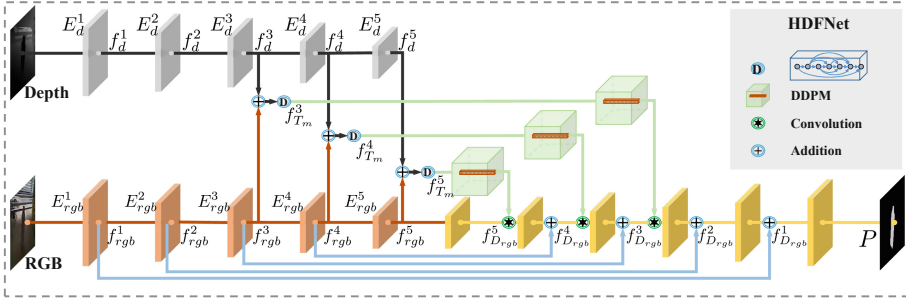
the repeated down-sampling lead to the loss of numerous details in the predicted results. Although many methods frequently combine shallower features to restore feature resolution, the improvement is still limited. While some approaches [9, 17] leverage CRF post-processing to refine subtle structures, which has a large computational cost. In this work, we design a new hybrid enhanced loss function (HEL). The HEL encourages the consistency between the area around edges and the interior of objects, thereby achieving sharper boundaries and a solid saliency area.

Our main contributions are summarized as follows:

- We propose a simple yet effective hierarchical dynamic filtering network (HDFNet) for RGB-D SOD. Especially, we provide a new perspective to utilize depth information. The depth and RGB features are combined to generate region-aware dynamic filters to guide the decoding in RGB stream.
- We propose a hybrid enhanced loss and verify its effectiveness in both RGB and RGB-D SOD tasks. It can effectively optimize the details of predictions and enhance the consistency of salient regions without additional parameters.
- We compare the proposed method with twelve state-of-the-art methods on eight datasets. It achieves the best performance under six evaluation metrics. Meanwhile, we implement a forward reasoning speed of 52 FPS on an NVIDIA GTX 1080 Ti GPU. The size of our VGG16-based model is about 170 MB (Fig. 1).

## 2 Related Word

**RGB-D Salient Object Detection.** The early methods are mainly based on hand-crafted features, such as contrast [6] and shape [7]. Limited by the representation ability of the features, they can not cope with complex scenes. Please refer to [12] for more details about traditional methods. In recent years, FCN-based methods have shown great potential and some of them achieve very good performance in the RGB-D SOD task [12, 36, 49]. Chen and Li [2] progressively combine the current depth/RGB features and the preceding fused feature by a series of convolution and element-wise addition operations to build the cross fusion modules. Recently, they concatenate depth and RGB features and feed them into an additional CNN stream to achieve multi-level cross-modal fusion [3]. Wang and Gong [42] respectively build a saliency prediction stream for RGB and depth inputs and then fuse their predictions and their preceding features to obtain final prediction via several convolutional layers. Zhao *et al* [49] insert a lightweight net between adjacent encoding blocks to compute a contrast map from the depth input and use it to enhance the features from the RGB stream. Piao *et al* [36] combine multi-level paired complementary features from RGB and depth streams by convolution and nonlinear operations. Fan *et al* [12] design a depth depurator to remove the low-quality depth input, and for high-quality one they feed the concatenated 4-channel input into a convolutional neural network to achieve cross-modal fusion. Different from these methods, we use the RGB-depth mixed features to generate “adaptive” multi-scale convolution kernels to filter and enhance the decoding features from the RGB stream.

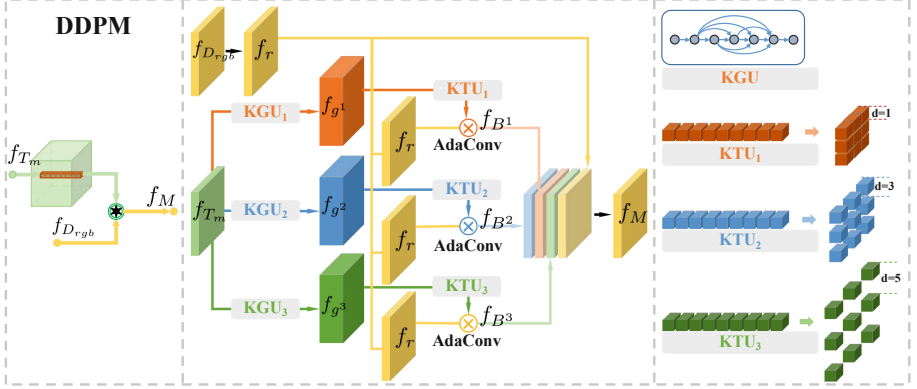


**Fig. 2.** The overall architecture of HDFNet. The network is based on two-stream structure. The two encoders use the same network (such as VGG-16 [40], VGG-19 [40], or ResNet-50 [16]), and are fed RGB and depth images, respectively. The details of HDFNet are introduced in Sect. 3.

**Dynamic Filters.** The works closely related to ours are [20] and [15]. The conception of the dynamic filter is firstly proposed in video and stereo prediction task [20]. The filter is utilized to enhance the representation of its corresponding input in a self-learning manner. While we use multi-modal information to generate multi-scale filters to dynamically strengthen the cross-modal complementarity and suppress the inter-modality incompatibility. Besides, the kernel computation in [20] introduces a large number of parameters and is difficultly extended at multiple scales, which significantly increases parameters and causes optimization difficulties. To efficiently achieve hierarchical dynamic filters, we introduce the idea of depth-wise separable convolution [18] and dilated convolution [48]. In [15], the filters are computed by pooling the input feature. They share kernel parameters across different positions, which is only an image-specific filter generator. In contrast, we design position-specific and image-specific filters to provide cross-modal contextual guidance for the decoder. The parameter update of dynamic filters is determined by the gradients of local neighborhoods to achieve more targeted adjustments and guarantee the overall performance of optimization.

### 3 Proposed Method

In this section, we first introduce the overall structure of the proposed method and then detail two main components, including the dynamic dilated pyramid module (DDPM) and the hybrid enhanced loss (HEL).



**Fig. 3.** The structure of the dynamic dilated pyramid module. The DDPM contains two submodules: kernel generation units (KGUs) and kernel transformation units (KTUs). KGUs generate adaptive kernel tensors and KTUs transform these tensors into the regular form of convolution kernels with different dilation rates.

### 3.1 Two Stream Structure

We build a two-stream network, which structure is shown in Fig. 2. It has two inputs: one is an RGB image and the other is a depth image, which corresponds to the RGB and depth streams, respectively. Through convolution blocks  $\{E_{rgb}^i\}_{i=1}^5$  and  $\{E_d^i\}_{i=1}^5$  in two encoding networks, we can obtain the intermediate features with different resolutions, which are recorded as  $f^1, f^2, f^3, f^4, f^5$  from large to small. The third-level features still retain enough valid information. Besides, the shallower features contain more noise and also cause higher computational cost due to the larger resolution. To balance efficiency and effectiveness, we only utilize the features  $f_d^3, f_d^4, f_d^5$  from the deepest three blocks in the depth stream. These features are respectively combined with the features  $f_{rgb}^3, f_{rgb}^4, f_{rgb}^5$  from the RGB stream. Then, we use a dense block [19] to build the transport layer, which combines rich and various receptive fields and generates powerful mixed features  $f_{T_m}$  with both spatial structures and appearance details. These features are fed into the DDPM to produce multi-scale convolution kernels that are used to filter the features  $f_{D_{rgb}}$  from the decoder. The resulted features  $f_M$  are merged in the top-down pathway by element-wise addition. After recovering the resolution layer by layer, we obtain the final prediction  $P$ , which is supervised by the ground truth  $G$ .

### 3.2 Dynamic Dilated Pyramid Module

In order to make more reasonable and effective use of the mixed features  $f_{T_m}$  from the dense transport layer, we employ DDPMs to generate the adaptive kernel for decoding RGB features. The DDPMs contain two inputs: the mixed feature  $f_{T_m}$  and the feature  $f_{D_{rgb}}$  from the decoder. On one hand, for specific

---

**Algorithm 1:** The operation process of adaptive convolution  $\otimes$  related to KTU $_j$  in DDPM $^i$ .

---

**Input:**  $f_r^i = \mathcal{R}(f_{D_{rgb}}^i) \in \mathbb{R}^{N \times C' \times H' \times W'}$ ,  $f_{g^j}^i \in \mathbb{R}^{N \times (9 \times C') \times H' \times W'}$

**Output:**  $f_{B^j}^i \in \mathbb{R}^{N \times C' \times H' \times W'}$ .

```

1  $d \leftarrow j \times 2 - 1$ ;
2 pad  $f_r^i$  with 0 from  $(H', W')$  to  $(H' + 2 \times d, W' + 2 \times d)$ ;
3 for  $n \leftarrow 0$  to  $N - 1$  do
4   for  $c \leftarrow 0$  to  $C' - 1$  do
5     for  $h \leftarrow d$  to  $H' + d - 1$  do
6       for  $w \leftarrow d$  to  $W' + d - 1$  do
7          $(f_{B^j}^i)_{[n,c,h,w]} \leftarrow \sum_{l=-1}^1 \sum_{m=-1}^1 \{ (f_{g^j}^i)_{[n,(l+1) \times 3 + (m+1),h,w]} \times (f_r^i)_{[n,c,h+l \times d,w+m \times d]} \}$ ;

```

---

position in feature maps  $f_{D_{rgb}}$ , we use kernel generation units (KGUs) to yield independent weight tensors, i.e.  $f_g$ , that can cover a  $3 \times 3$ ,  $7 \times 7$  or  $11 \times 11$  square neighborhood. KGUs are also a kind of dense structure [19]. The module contains 4 densely connected layers and each layer is connected to all the others in a feed-forward fashion, which can further strengthen feature propagation and expression capabilities, encourage feature reuse and greatly improve parameter efficiency. Then, by recombining kernel tensors and inserting different numbers of zeros, kernel transformation units (KTUs) construct regular convolution kernels with different dilation rates. Please see ‘‘KTU’’ shown in Fig. 3 and introduced in Algorithm 1 for a more intuitive presentation. On the other hand, after preliminary dimension reduction, the other input  $f_{D_{rgb}}$  is re-weighted and integrated into three parallel branches to obtain the enhanced features  $\{f_{B^j}\}_{j=1}^3$ . Note that this is actually a channel-wise adjustment and the operation of each channel is independent. Finally, after concatenating and merging  $\{f_{B^j}\}_{j=1}^3$  and the reduced  $f_{D_{rgb}}$ , the resulted features  $\{f_M^i\}_{i=3}^5$  become more discriminative.

The entire process can be formulated as follows:

$$\begin{aligned}
f_M^i &= \text{DDPM}^i(f_{D_{rgb}}^i, f_{T_m}^i) \\
&= \mathcal{F}(\mathcal{C}(\mathcal{R}(f_{D_{rgb}}^i), f_{B^1}^i, f_{B^2}^i, f_{B^3}^i)) \\
&= \mathcal{F}(\mathcal{C}(\mathcal{R}(f_{D_{rgb}}^i), \mathcal{KTU}_1^i(\mathcal{KGU}_1^i(f_{T_m}^i)) \otimes \mathcal{R}(f_{D_{rgb}}^i), \\
&\quad \mathcal{KTU}_2^i(\mathcal{KGU}_2^i(f_{T_m}^i)) \otimes \mathcal{R}(f_{D_{rgb}}^i), \\
&\quad \mathcal{KTU}_3^i(\mathcal{KGU}_3^i(f_{T_m}^i)) \otimes \mathcal{R}(f_{D_{rgb}}^i))),
\end{aligned} \tag{1}$$

where  $f_M^i$  represents the feature from the DDPM $^i$  related to the  $f_{D_{rgb}}^i$ .  $\text{DDPM}(\cdot)$ ,  $\mathcal{KGU}(\cdot)$  and  $\mathcal{KTU}(\cdot)$  denote the operation of the corresponding module.  $\mathcal{R}(\cdot)$  is a  $1 \times 1$  convolution operation, which is used to reduce the number of channels from 64 to 16.  $\otimes$  is an adaptive convolution operation as shown in Algorithm 1.  $\mathcal{C}(\cdot)$  is a concatenation operation and  $\mathcal{F}(\cdot)$  is a  $3 \times 3$  convolution to

fuse the concatenated features from different branches. More details is as shown in Fig. 3.

### 3.3 Hybrid Enhanced Loss

No matter for RGB or RGB-D based SOD tasks, good prediction requires the salient area to be clearly and completely highlighted. This contains two aspects: one is the sharpness of boundaries and the other is the consistency of intra-class. We start with the loss function and design a new loss to constrain the edges and the fore-/background regions to separately achieve high-contrast predictions.

The common loss function in the SOD task is binary cross entropy (BCE). It is a pixel-level loss, which independently performs error calculation and supervision at different positions. The main form is as follows:

$$L_{bce} = \frac{1}{N \times H \times W} \sum_n^N \sum_h^H \sum_w^W [g \log p + (1 - g) \log(1 - p)], \quad (2)$$

where  $P = \{p | 0 < p < 1\} \in \mathbb{R}^{N \times 1 \times H \times W}$  and  $G = \{g | 0 < g < 1\} \in \mathbb{R}^{N \times 1 \times H \times W}$  respectively represent the prediction and the corresponding ground truth.  $N$ ,  $H$  and  $W$  are the batchsize, height and width of the input data, respectively. It calculates the error between the ground truth  $g$  and the prediction  $p$  at each position, and the loss  $L_{bce}$  accumulates and averages the errors of all positions.

In order to further enhance the strength of supervision at higher levels such as edges and regions, we specially constrain and optimize the regions near the edges. In particular, the loss is formulated as follows:

$$L_e = \frac{\sum_h^H \sum_w^W (e * |p - g|)}{\sum_h^H \sum_w^W e}, \quad (3)$$

$$e = \begin{cases} 0 & \text{if } (G - \mathcal{P}(G))_{[h,w]} = 0, \\ 1 & \text{if } (G - \mathcal{P}(G))_{[h,w]} \neq 0, \end{cases}$$

where  $L_e$  represents the edge enhanced loss (EEL), and  $\mathcal{P}(\cdot)$  denotes the average pooling operation with a  $5 \times 5$  slide window. In Eq. 3, we can obtain the local region near the contour of the ground truth by calculating  $e$ . In this region, the difference  $L_e$  between the prediction  $p$  and the ground truth  $g$  can be calculated. Through this loss, the optimization process can target the contours of salient objects.

In addition, we also design a region enhanced loss (REL) to constrain the prediction of intra-class. By respectively calculating the prediction errors within the foreground class and the background class, fore-/background predictions can be independently optimized. Specifically, the REL  $L_r$  is written as:

$$\begin{aligned}
L_r &= \frac{\sum_n (L_f + L_b)}{N}, \\
L_f &= \frac{\sum_h \sum_w (g - g * p)}{\sum_h \sum_w g}, \\
L_b &= \frac{\sum_h \sum_w (1 - g) * p}{\sum_h \sum_w (1 - g)},
\end{aligned} \tag{4}$$

where  $L_f$  and  $L_b$  denote the fore-/background losses, respectively. The losses compute the normalized prediction errors in the intra-class regions. They depict the region-level supervision. Finally, we integrate these three losses ( $L_{bce}$ ,  $L_e$  and  $L_r$ ) to obtain the hybrid enhanced loss (HEL), which can optimize the prediction at two different levels. The total loss is expressed as follows:

$$L = L_{bce} + L_e + L_r. \tag{5}$$

## 4 Experiments

### 4.1 Datasets

To fully verify the effectiveness of the proposed method, we evaluated the results on eight benchmark datasets. **LFS**D [24] is a small dataset that contains 100 images with depth information and human-labeled ground truths and is built for saliency detection on the light field. **NJUD** [21] contains 1,985 groups of RGB, depth, and label images, which are collected from the Internet, 3D movies, and photographs taken by a Fuji W3 stereo camera. **NLPR** [34] is also called **RGBD1000**, which contains 1,000 natural RGBD images captured by Microsoft Kinect together with the human-marked ground truth. **RGBD135** [6] is also named **DES**, which consists of 135 images about indoor scenes collected by Microsoft Kinect. **SIP** [12] includes 1,000 images with many challenging situations from various outdoor scenarios and these images emphasize salient persons in real-world scenes. **SSD** [52] contains 80 images picked up from three stereo movies. **STEREO** [32] is also called **SSB**, which contains 1,000 stereoscopic images downloaded from the Internet. **DUTRGBD** [36] is a new and large dataset and contains 800 indoor and 400 outdoor scenes paired with the depth maps and ground truths.

For comprehensively and fairly evaluating different methods, we follow the setting of [36]. On the DUTRGBD, we use 800 images for training and 400 images for testing. For the other seven datasets, we follow the data partition of [2, 4, 14, 36] to use 1,485 samples from the NJUD and 700 samples from the NLPR as the training set and the remaining samples in these datasets are used for testing.

### 4.2 Evaluation Metrics

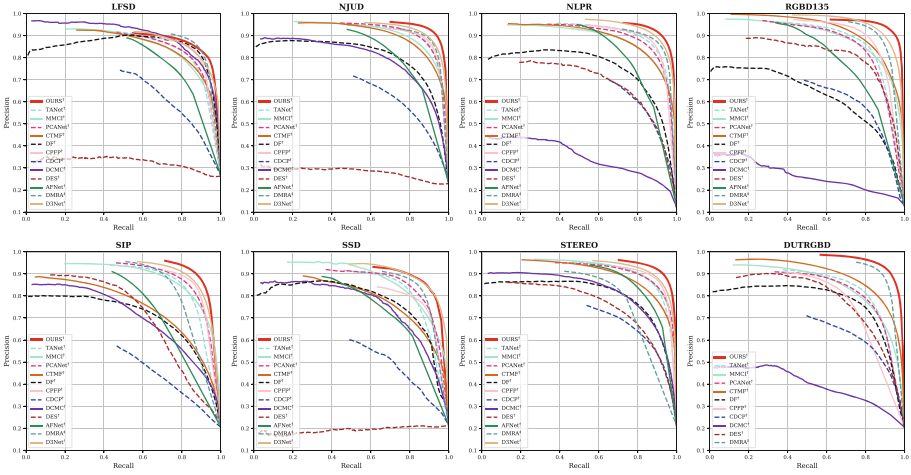
There are six widely used metrics for evaluating RGB and RGB-D SOD models: Precision-Recall (PR) curve, F-measure [1], weighted F-measure [31], MAE [35],



**Table 1.** Results ( $\uparrow$ :  $F_{max}$ ,  $F_{ada}$  [1],  $F_{\beta}^{\omega}$  [31],  $S_m$  [10] and  $E_m$  [11];  $\downarrow$ : MAE [35]) of different RGB-D SOD methods across eight datasets. The best results are highlight in **red**.  $\ddagger$ : Traditional methods.  $\dagger$ : VGG-16 [40] as backbone.  $\ddagger$ : VGG-19 [40] as backbone.  $\ddagger$ : ResNet-50 [16] as backbone. -: No data available.

	Metric	DESP <sub>14</sub> [6]	DCMC <sub>16</sub> [8]	CDCP <sub>17</sub> [54]	DF <sub>17</sub> [37]	CTMP <sub>14</sub> [14]	PCANet <sub>14</sub> [2]	MMCT <sub>15</sub> [4]	TANet <sub>15</sub> [3]	AFNet <sub>15</sub> [42]	CFPP <sub>15</sub> [49]	OURS <sup>†</sup>	DMRA <sub>15</sub> [36]	OURS <sup>‡</sup>	DdNet <sub>15</sub> [12]	OURS <sup>‡</sup>	
LFSO [24]	$F_{max}$	0.377	0.850	0.680	0.854	0.815	0.829	0.813	0.827	0.780	0.856	<b>0.860</b>	<b>0.872</b>	0.853	0.849	<b>0.883</b>	
	$F_{ada}$	0.227	0.815	0.634	0.810	0.781	0.793	0.779	0.794	0.742	0.813	<b>0.831</b>	<b>0.849</b>	0.833	0.801	<b>0.843</b>	
	$F_{\beta}^{\omega}$	0.274	0.601	0.518	0.642	0.696	0.716	0.663	0.719	0.671	0.775	<b>0.792</b>	<b>0.811</b>	0.793	0.756	<b>0.806</b>	
	MAE	0.416	0.155	0.199	0.142	0.200	0.112	0.132	0.111	0.133	0.088	<b>0.085</b>	<b>0.076</b>	0.083	0.099	<b>0.076</b>	
	$S_m$	0.440	0.754	0.658	0.786	0.796	0.800	0.787	0.801	0.738	0.828	<b>0.847</b>	<b>0.847</b>	0.841	0.832	<b>0.854</b>	
	$E_m$	0.492	0.842	0.737	0.841	0.851	0.856	0.840	0.851	0.810	0.867	<b>0.883</b>	<b>0.899</b>	0.886	0.860	<b>0.891</b>	
NIJD [21]	$F_{max}$	0.328	0.769	0.661	0.789	0.857	0.887	0.868	0.888	0.804	0.890	<b>0.924</b>	0.896	<b>0.922</b>	0.903	<b>0.922</b>	
	$F_{ada}$	0.165	0.715	0.618	0.744	0.788	0.834	0.813	0.844	0.768	0.837	<b>0.894</b>	0.872	<b>0.887</b>	0.840	<b>0.889</b>	
	$F_{\beta}^{\omega}$	0.234	0.497	0.510	0.545	0.720	0.803	0.739	0.805	0.696	0.828	<b>0.881</b>	<b>0.849</b>	<b>0.877</b>	0.833	<b>0.877</b>	
	MAE	0.448	0.167	0.182	0.151	0.085	0.059	0.079	0.061	0.100	0.053	<b>0.037</b>	0.051	<b>0.038</b>	0.051	<b>0.038</b>	
	$S_m$	0.413	0.703	0.672	0.735	0.849	0.877	0.878	0.872	0.772	0.878	<b>0.911</b>	0.885	<b>0.911</b>	0.895	<b>0.908</b>	
	$E_m$	0.491	0.796	0.751	0.818	0.866	0.909	0.882	0.909	0.847	0.900	<b>0.934</b>	0.920	<b>0.932</b>	0.901	<b>0.932</b>	
NLPR [34]	$F_{max}$	0.695	0.413	0.687	0.752	0.841	0.864	0.841	0.876	0.816	0.883	<b>0.917</b>	0.888	<b>0.919</b>	0.904	<b>0.927</b>	
	$F_{ada}$	0.583	0.328	0.591	0.683	0.724	0.795	0.730	0.796	0.747	0.818	<b>0.878</b>	0.855	<b>0.883</b>	0.834	<b>0.889</b>	
	$F_{\beta}^{\omega}$	0.254	0.259	0.501	0.516	0.679	0.762	0.676	0.780	0.693	0.807	<b>0.869</b>	0.839	<b>0.871</b>	0.826	<b>0.882</b>	
	MAE	0.300	0.196	0.114	0.100	0.051	0.044	0.059	0.041	0.058	0.037	<b>0.031</b>	0.031	<b>0.027</b>	0.034	<b>0.023</b>	
	$S_m$	0.582	0.550	0.724	0.769	0.860	0.873	0.856	0.886	0.799	0.884	<b>0.916</b>	0.898	<b>0.915</b>	0.906	<b>0.923</b>	
	$E_m$	0.760	0.685	0.786	0.840	0.869	0.916	0.872	0.916	0.884	0.920	<b>0.948</b>	0.942	<b>0.951</b>	0.934	<b>0.957</b>	
RGBD135 [6]	$F_{max}$	0.800	0.311	0.651	0.625	0.865	0.842	0.839	0.853	0.775	0.882	<b>0.934</b>	0.906	<b>0.941</b>	0.917	<b>0.932</b>	
	$F_{ada}$	0.695	0.234	0.594	0.573	0.778	0.774	0.762	0.795	0.730	0.829	<b>0.919</b>	0.867	<b>0.918</b>	0.876	<b>0.912</b>	
	$F_{\beta}^{\omega}$	0.301	0.169	0.478	0.392	0.686	0.711	0.650	0.740	0.641	0.787	<b>0.902</b>	0.843	<b>0.913</b>	0.831	<b>0.895</b>	
	MAE	0.288	0.196	0.120	0.131	0.055	0.050	0.065	0.046	0.068	0.038	<b>0.020</b>	0.030	<b>0.017</b>	0.030	<b>0.021</b>	
	$S_m$	0.632	0.449	0.709	0.685	0.863	0.843	0.848	0.858	0.770	0.872	<b>0.932</b>	0.899	<b>0.927</b>	0.904	<b>0.926</b>	
	$E_m$	0.817	0.676	0.810	0.806	0.911	0.912	0.904	0.919	0.874	0.927	<b>0.973</b>	0.944	<b>0.976</b>	0.956	<b>0.974</b>	
SIP [12]	$F_{max}$	0.720	0.680	0.544	0.704	0.720	0.860	0.840	0.851	0.756	0.870	<b>0.904</b>	0.847	<b>0.907</b>	0.882	<b>0.910</b>	
	$F_{ada}$	0.644	0.645	0.495	0.673	0.684	0.825	0.795	0.809	0.705	0.819	<b>0.863</b>	0.815	<b>0.870</b>	0.831	<b>0.875</b>	
	$F_{\beta}^{\omega}$	0.342	0.413	0.397	0.406	0.535	0.768	0.711	0.748	0.617	0.788	<b>0.835</b>	0.734	<b>0.844</b>	0.793	<b>0.848</b>	
	MAE	0.298	0.186	0.224	0.185	0.139	0.071	0.086	0.075	0.118	0.064	<b>0.050</b>	0.088	<b>0.047</b>	0.063	<b>0.047</b>	
	$S_m$	0.616	0.683	0.595	0.653	0.716	0.842	0.833	0.835	0.720	0.850	<b>0.878</b>	0.800	<b>0.885</b>	0.864	<b>0.886</b>	
	$E_m$	0.751	0.786	0.722	0.794	0.824	0.900	0.886	0.894	0.815	0.899	<b>0.920</b>	0.858	<b>0.924</b>	0.903	<b>0.924</b>	
SSD [32]	$F_{max}$	0.260	0.750	0.576	0.763	0.755	0.844	0.823	0.834	0.735	0.801	<b>0.872</b>	0.858	<b>0.883</b>	0.872	<b>0.885</b>	
	$F_{ada}$	0.163	0.684	0.521	0.709	0.709	0.786	0.748	0.766	0.694	0.726	<b>0.844</b>	0.821	<b>0.847</b>	0.793	<b>0.842</b>	
	$F_{\beta}^{\omega}$	0.172	0.480	0.429	0.536	0.622	0.733	0.662	0.727	0.589	0.708	<b>0.808</b>	0.787	<b>0.819</b>	0.780	<b>0.821</b>	
	MAE	0.500	0.168	0.219	0.151	0.100	0.063	0.082	0.063	0.118	0.082	<b>0.048</b>	0.058	<b>0.046</b>	0.058	<b>0.045</b>	
	$S_m$	0.341	0.706	0.603	0.741	0.776	0.842	0.813	0.839	0.714	0.807	<b>0.866</b>	0.856	<b>0.875</b>	0.866	<b>0.879</b>	
	$E_m$	0.475	0.790	0.714	0.801	0.838	0.890	0.860	0.886	0.803	0.832	<b>0.913</b>	0.898	<b>0.911</b>	0.892	<b>0.911</b>	
STEREO [32]	$F_{max}$	0.738	0.789	0.704	0.789	0.848	0.875	0.877	0.878	0.848	0.889	<b>0.918</b>	0.802	<b>0.916</b>	0.897	<b>0.910</b>	
	$F_{ada}$	0.594	0.742	0.666	0.742	0.771	0.826	0.829	0.835	0.807	0.830	<b>0.879</b>	0.762	<b>0.875</b>	0.833	<b>0.867</b>	
	$F_{\beta}^{\omega}$	0.375	0.520	0.358	0.549	0.698	0.778	0.769	0.787	0.752	0.817	<b>0.863</b>	0.647	<b>0.859</b>	0.815	<b>0.853</b>	
	MAE	0.295	0.148	0.149	0.141	0.086	0.064	0.068	0.060	0.075	0.051	<b>0.039</b>	0.087	<b>0.040</b>	0.054	<b>0.041</b>	
	$S_m$	0.642	0.731	0.713	0.757	0.848	0.875	0.873	0.871	0.825	0.879	<b>0.906</b>	0.752	<b>0.903</b>	0.891	<b>0.900</b>	
	$E_m$	0.696	0.831	0.796	0.888	0.870	0.907	0.905	0.916	0.887	0.907	<b>0.937</b>	0.816	<b>0.934</b>	0.911	<b>0.931</b>	
DUTRGBD [36]	$F_{max}$	0.770	0.444	0.658	0.744	0.842	0.809	0.804	0.823	-	0.787	<b>0.926</b>	0.908	<b>0.934</b>	-	<b>0.930</b>	
	$F_{ada}$	0.667	0.405	0.633	0.747	0.792	0.760	0.753	0.778	-	0.735	<b>0.892</b>	0.883	<b>0.894</b>	-	<b>0.885</b>	
	$F_{\beta}^{\omega}$	0.380	0.284	0.521	0.536	0.682	0.688	0.628	0.705	-	0.638	<b>0.865</b>	0.852	<b>0.871</b>	-	<b>0.864</b>	
	MAE	0.280	0.243	0.159	0.145	0.097	0.100	0.112	0.093	-	0.100	<b>0.040</b>	0.048	<b>0.039</b>	-	<b>0.041</b>	
	$S_m$	0.659	0.499	0.687	0.729	0.831	0.801	0.791	0.808	-	0.749	<b>0.905</b>	0.887	<b>0.911</b>	-	<b>0.907</b>	
	$E_m$	0.751	0.712	0.794	0.842	0.882	0.863	0.856	0.871	-	0.815	<b>0.938</b>	0.930	<b>0.941</b>	-	<b>0.938</b>	
AveMetric	$F_{max}$	0.654	0.666	0.642	0.756	0.811	0.861	0.850	0.862	0.801	0.868	<b>0.914</b>	0.855	<b>0.915</b>	0.893	<b>0.915</b>	
	$F_{ada}$	0.534	0.618	0.595	0.714	0.747	0.814	0.794	0.815	0.755	0.813	<b>0.878</b>	0.822	<b>0.878</b>	0.834	<b>0.877</b>	
	$F_{\beta}^{\omega}$	0.325	0.425	0.491	0.502	0.652	0.761	0.712	0.712	0.764	0.684	0.784	<b>0.857</b>	0.756	<b>0.859</b>	0.810	<b>0.858</b>
	MAE	0.325	0.179	0.174	0.151	0.099	0.068	0.081	0.067	0.093	0.061	<b>0.041</b>	0.069	<b>0.041</b>	0.055	<b>0.041</b>	
	$S_m$	0.585	0.661	0.669	0.721	0.809	0.853	0.844	0.853	0.773	0.853	<b>0.898</b>	0.824	<b>0.900</b>	0.883	<b>0.899</b>	
	$E_m$	0.686	0.781	0.765	0.822	0.859	0.899	0.885	0.901	0.853	0.892	<b>0.933</b>	0.876	<b>0.933</b>	0.909	<b>0.932</b>	

S-measure [10] and E-measure [11]. **PR Curve.** We use a series of fixed thresholds from 0 to 255 to binarize the gray prediction map, and then calculate several groups of precision ( $Pre$ ) and recall ( $Rec$ ) with ground truth by  $Pre = \frac{TP}{TP+FP}$  and  $Rec = \frac{TP}{TP+FN}$ . Based on them, we can plot a precision-recall curve to describe the performance of the model. **F-measure** [1]. It is a region-based similarity metric and is formulated as the weighted harmonic mean (the weight is set to 0.3) of  $Pre$  and  $Rec$ . In this paper, we employ the threshold filtering from 0 to 255 to get  $F_{max}$ , and use twice the mean value of the prediction  $P$  as the threshold to obtain  $F_{ada}$ . In addition, since F-measure reflects the performance of the binary predictions under different thresholds, we evaluate the consistency and uniformity at the regional level according to F-measure threshold curves. **weighted F-measure** ( $F_{\beta}^{\omega}$ ) [31]. It is proposed to improve the existing metric F-measure. It defines a weighted precision, which is a measure of exactness, and a weighted recall, which is a measure of completeness and follows the



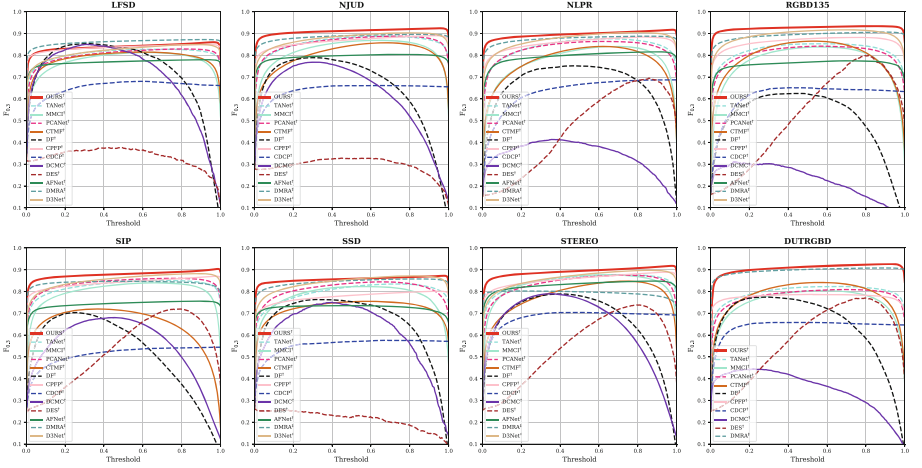
**Fig. 4.** Precision (vertical axis) recall (horizontal axis) curves on eight RGB-D salient object detection datasets.

form of F-measure. **MAE** [35]. This metric estimates the approximation degree between the saliency map and ground-truth map, and it is normalized to  $[0, 1]$ . It focuses on pixel-level performance. **S-measure** ( $S_m$ ) [10]. It calculates the object-/region-aware structure similarities  $S_o/S_r$  between prediction and ground truth by the equation:  $S_m = \alpha \cdot S_o + (1 - \alpha) \cdot S_r$ ,  $\alpha = 0.5$ . **E-measure** ( $E_m$ ) [11]. This measure utilizes the mean-removed predictions and ground truths to compute the similarity, which characterizes both image-level statistics and local pixel matching.

### 4.3 Implementation Details

**Parameter Setting.** Two encoders of the proposed model are based on the same model, such as VGG-16 [40], VGG-19 [40], and ResNet-50 [16]. In both encoders, only the convolutional layers in corresponding classification networks are retained, and the last pooling layer of VGG-16 and VGG-19 is removed at the same time. During the training phase, we use the weight parameters pre-trained on the ImageNet to initialize the encoders. Also, since the depth image is a single channel data, we change the channel number of its corresponding input layer from 3 to 1, and its parameters are initialized randomly by PyTorch. The parameters of the remaining structures are all initialized randomly.

**Training Setting.** During the training stage, we apply random horizontal flipping, random rotating as data augmentation for RGB images and depth images. In addition, we employ random color jittering and normalization for RGB images. We use the momentum SGD optimizer with a weight decay of  $5e-4$ , an initial learning rate of  $5e-3$ , and a momentum of 0.9. Besides, we apply a “poly” strategy [28] with a factor of 0.9. The input images are resized



**Fig. 5.** F-measure (vertical axis) threshold (horizontal axis) curves on eight RGB-D salient object detection datasets.

to  $320 \times 320$ . We train the model for 30 epochs on an NVIDIA GTX 1080 Ti GPU with a batch size of 4 to obtain the final model.

**Testing Details.** During the testing stage, we resize RGB and depth images to  $320 \times 320$  and normalize RGB images. Besides, the final prediction is rescaled to the original size for evaluation.

#### 4.4 Comparisons

In order to fully demonstrate the effectiveness of the proposed method, we compared it with the existing twelve RGB-D based SOD models, including DES [6], DCMC [8], CDCP [53], DF [37], CTMF [14], PCANet [2], MMCI [4], TANet [3], AFNet [42], CFPF [49], DMRA [36] and D3Net [12]. For fair comparisons, all saliency maps of these methods are directly provided by authors or computed by their released codes. Besides, the codes and results of AFNet [42] and D3Net [12] on the DUTRGBD [36] dataset are not publicly available. Therefore, their results on this dataset are not listed.

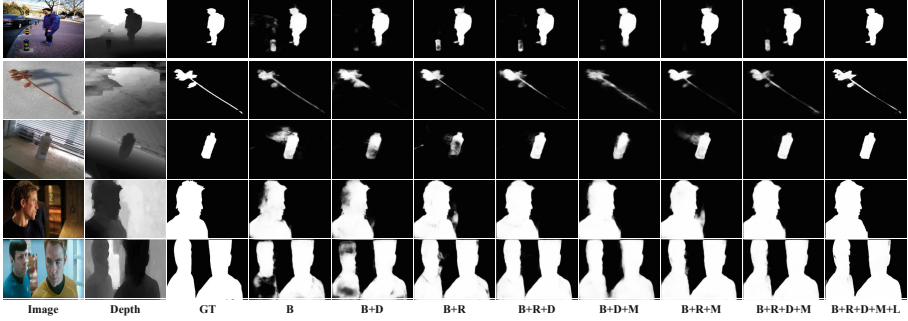
**Quantitative Evaluation.** In Table 1, we list the results of all competitors on eight datasets and six metrics. It can be seen that the proposed method performs best on most datasets and achieve significant performance improvement. On the DUTRGBD [36], our models based on VGG-16, VGG-19 and ResNet-50 have surpassed the second-best model DMRA [36] by 2.02%, 2.85% and 2.45% on  $F_{max}$ , and 16.09%, 17.88% and 13.56% on MAE. At the same time, on the recent dataset SIP [12], they have increased by 3.83%, 4.65% and 5.23% on  $F_{ada}$ , 5.22%, 6.37% and 6.84% on  $F_{\beta}^w$ , and 20.94%, 24.65% and 24.91% on MAE, over the D3Net [12]. Because the existing RGB-D SOD datasets are relatively small, we propose a new calculation method to measure the performance of



**Fig. 6.** The visualization results of some recent methods and ours.

models. According to the proportion of each testing set in all testing datasets, the results on all datasets are weighted and summed to obtain an overall performance evaluation, which is listed in the row “AveMetric” in Table 1. It can be seen that our structure achieves similar and excellent results on different backbones, which shows that our structure has less dependence on the performance of the backbone. In addition, we show a scatter plot based on the average performance of each model on all datasets and the model size in Fig. 1. Our model has the smallest size while achieving the best result. We demonstrate the PR curves and the F-measure curves in Fig. 4 and Fig. 5. Our approach (red solid line) achieves very good results on these datasets. As shown in Fig. 5, our results are much flatter at most thresholds, which reflects that our prediction results are more uniform and consistent.

**Qualitative Evaluation.** In Fig. 6, we list some representative results. These examples include scenarios with varying complexity, as well as different types of objects, including cluttered background (Column 1 and 2), simple scene (Column 3 and 4), small objects (Column 5), complex objects (Column 6 and 7), large objects (Column 8), multiple objects (Column 9 and 10) and low contrast between foreground and background (Column 11 and 12). It can be seen that the proposed method can consistently produce more accurate and complete saliency maps with higher contrast.



**Fig. 7.** Visual comparisons for showing the benefits of the proposed components. GT: Ground truth; B: Baseline; D: Dense transport layer for depth features; R: Dense transport layer for RGB features; M: DDPM; L: HEL.

#### 4.5 Ablation Study

In this section, we perform ablation analysis over the main components of the HDFNet and further investigate their importance and contributions. Our baseline model, i.e. Model 1, uses the commonly used encoder-decoder structure, and all ablation experiments are based on the VGG-16 backbone. In the baseline model, the output features of the last three stages in the depth stream are added to the decoder after compressing the channel to 64 through an independent  $1 \times 1$  convolution. In order to evaluate the benefits of cross-modal fusion at the dense transport layer (i.e. Model 6), we feed single-modal features into this layer to build Model 2 (i.e. “ $+T_d$ ”) and Model 4 (i.e. “ $+T_{rgb}$ ”). Thus, the followed dynamic filters in the DDPM will be determined only by depth features or RGB features, respectively.

**Dynamic Dilated Pyramid Module.** Based on Model 2, Model 4, and Model 6, we add the dynamic dilated pyramid module to obtain Model 3, Model 5, and Model 7, respectively. In Table 2, we show the performance improvement contributed by different structures in terms of the weighted average metrics “AveMetric”. It can be seen that the DDPM significantly improves performance. Specifically, by comparing Model 3, 5 and 7 with Model 2, 4 and 6, we achieve a relative improvement of 1.47%, 3.11% and 2.11% in terms of  $F_\beta^\omega$  and 5.01%, 10.29% and 6.77% in terms of MAE, respectively. We can see that even without the HEL, the average performance of Model 7 already exceeds these existing models. More comparisons can be found in the supplementary material.

In addition, we compare the design of the dynamic filter in DCM [15] with ours. It can be seen that the proposed DDPM (Model 7) has obvious advantages over the DCM (Model 8), and it respectively increases by 3.91%, 5.60%, and 18.07% in terms of  $F_{ada}$ ,  $F_\beta^\omega$  and MAE. In Fig. 7, we can see that the noise in depth images interferes with the final predictions. By the cross-modal guidance from the DDPMs, the interference is effectively suppressed.

**Table 2.** Ablation experiments. + $\mathbf{T}_d$ : Using a dense transport layer for depth features. + $\mathbf{T}_{rgb}$ : Using a dense transport layer for RGB features. +**DDPM**: Using a DDPM after the transport layer. +**DCM**: Using the DCM [15] after the transport layer. + $\mathbf{L}_e$ : Using the edge loss as the auxiliary loss. + $\mathbf{L}_f$ : Using the foreground loss as the auxiliary loss. + $\mathbf{L}_b$ : Using the background loss as the auxiliary loss.

Model	No.	Baseline	+ $\mathbf{T}_d$	+ $\mathbf{T}_{rgb}$	+DDPM	+DCM	+ $\mathbf{L}_e$	+ $\mathbf{L}_f$	+ $\mathbf{L}_b$	$F_{max}$	$F_{ada}$	$F_{\beta}^{\circ}$	MAE	$S_m$	$E_m$
Ours <sup>†</sup>	1	✓								0.875	0.819	0.768	0.067	0.865	0.898
	2	✓	✓							0.879	0.820	0.768	0.066	0.868	0.899
	3	✓	✓		✓					0.882	0.820	0.780	0.063	0.873	0.900
	4	✓		✓						0.884	0.839	0.787	0.060	0.874	0.909
	5	✓		✓	✓					0.896	0.852	0.811	0.054	0.886	0.916
	6	✓	✓	✓						0.898	0.846	0.803	0.056	0.884	0.913
	7	✓	✓	✓	✓					0.904	0.856	0.820	0.052	0.893	0.918
	8	✓	✓	✓		✓				0.878	0.823	0.777	0.064	0.871	0.903
	9	✓	✓	✓	✓		✓			0.909	0.878	0.849	0.044	0.898	0.929
	10	✓	✓	✓	✓			✓		0.909	0.845	0.827	0.050	0.887	0.916
	11	✓	✓	✓	✓				✓	0.907	0.874	0.836	0.048	0.895	0.926
	12	✓	✓	✓	✓		✓	✓	✓	0.914	0.878	0.857	0.041	0.898	0.933
R3Net <sub>18</sub> [9]	13									0.828	0.714	0.716	0.072	0.831	0.830
	14						✓	✓	✓	0.832	0.731	0.740	0.069	0.835	0.844
CPD <sub>19</sub> [45]	15									0.848	0.790	0.769	0.052	0.856	0.889
	16						✓	✓	✓	0.849	0.804	0.792	0.049	0.857	0.898
PoolNet <sub>19</sub> [26]	15									0.832	0.755	0.728	0.060	0.841	0.865
	16						✓	✓	✓	0.861	0.811	0.799	0.046	0.862	0.902
GCPANet <sub>20</sub> [5]	17									0.847	0.766	0.744	0.061	0.854	0.869
	18						✓	✓	✓	0.854	0.779	0.773	0.055	0.856	0.880

**Hybrid Enhanced Loss.** As shown in Table 2, the proposed hybrid enhanced loss brings huge performance improvements by comparing Model 7 with Model 12. We evaluate each component in the HEL (Model 9, 10, and 11) and all of them contribute to the final performance. In addition, the benefits of this loss are also clearly reflected in Fig. 5 where the curves of the proposed model are more straight, and Fig. 7 where the predictions of the model “B+R+D+M+L” have higher contrast than ones of the model “B+D+R+M”. Since the design goal of the HEL is to solve the general requirements of SOD tasks, we evaluate its effectiveness on several recent RGB SOD models [5, 9, 26, 45]. For a fair comparison, we retrain these models using the released code. Most of hyper-parameters are the same as the default values given by their corresponding code. The average performance “AveMetric” on five main RGB SOD datasets (DUTS [41], ECSSD [46], HKU-IS [23], PASCAL-S [25] and DUT-OMRON [47]) is shown in Table 2. More experimental details and results can be found in supplementary materials.

## 5 Conclusions

In this paper, we revisit the role that depth information should play in the RGB-D based SOD task. We consider the characteristics of spatial structures contained in depth information and combine it with RGB information with

rich appearance details. After that, the model generates adaptive filters with different receptive field sizes through the dynamic dilated pyramid module. It can make full use of semantic cues from multi-modal mixed features to achieve multi-scale cross-modal guidance, thereby enhancing the representation capabilities of the decoder. At the same time, we can obtain clearer predictions with the aid of additional region-level supervision to the regions around the edges and fore-/background regions. Extensive experiments on eight datasets and six metrics demonstrate the effectiveness of the designed components. The proposed approach achieves state-of-the-art performance with small model size and high running speed.

**Acknowledgements.** This work was supported in part by the National Key R&D Program of China #2018AAA0102003, National Natural Science Foundation of China #61876202, #61725202, #61751212 and #61829102, the Dalian Science and Technology Innovation Foundation #2019J12GX039, and the Fundamental Research Funds for the Central Universities #DUT20ZD212.

## References

1. Achanta, R., Hemami, S., Estrada, F., Süsstrunk, S.: Frequency-tuned salient region detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 1597–1604 (2009)
2. Chen, H., Li, Y.: Progressively complementarity-aware fusion network for RGB-D salient object detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 3051–3060 (2018)
3. Chen, H., Li, Y.: Three-stream attention-aware network for RGB-D salient object detection. *IEEE Trans. Image Process.* **28**(6), 2825–2835 (2019)
4. Chen, H., Li, Y., Su, D.: Multi-modal fusion network with multi-scale multi-path and cross-modal interactions for RGB-D salient object detection. *Pattern Recogn.* **86**, 376–385 (2019)
5. Chen, Z., Xu, Q., Cong, R., Huang, Q.: Global context-aware progressive aggregation network for salient object detection. In: AAAI Conference on Artificial Intelligence (2020)
6. Cheng, Y., Fu, H., Wei, X., Xiao, J., Cao, X.: Depth enhanced saliency detection method. In: Proceedings of the International Conference on Internet Multimedia Computing and Service, pp. 23–27 (2014)
7. Ciptadi, A., Hermans, T., Rehg, J.M.: An in depth view of saliency (2013)
8. Cong, R., Lei, J., Zhang, C., Huang, Q., Cao, X., Hou, C.: Saliency detection for stereoscopic images based on depth confidence analysis and multiple cues fusion. *IEEE Signal Process. Lett.* **23**(6), 819–823 (2016)
9. Deng, Z., et al.: R3Net: recurrent residual refinement network for saliency detection. In: International Joint Conference on Artificial Intelligence, pp. 684–690 (2018)
10. Fan, D.P., Cheng, M.M., Liu, Y., Li, T., Borji, A.: Structure-measure: a new way to evaluate foreground maps. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4548–4557 (2017)
11. Fan, D.P., Gong, C., Cao, Y., Ren, B., Cheng, M.M., Borji, A.: Enhanced-alignment measure for binary foreground map evaluation. In: International Joint Conference on Artificial Intelligence, pp. 698–704 (2018)

12. Fan, D.P., et al.: Rethinking RGB-D salient object detection: models, datasets, and large-scale benchmarks. arXiv preprint [arXiv:1907.06781](https://arxiv.org/abs/1907.06781) (2019)
13. Fan, D.P., Wang, W., Cheng, M.M., Shen, J.: Shifting more attention to video salient object detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 8554–8564 (2019)
14. Han, J., Chen, H., Liu, N., Yan, C., Li, X.: CNNs-based RGB-D saliency detection via cross-view transfer and multiview fusion. *IEEE Trans. Cybern.* **48**(11), 3171–3183 (2017)
15. He, J., Deng, Z., Qiao, Y.: Dynamic multi-scale filters for semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3562–3572 (2019)
16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
17. Hou, Q., Cheng, M.M., Hu, X., Borji, A., Tu, Z., Torr, P.H.: Deeply supervised salient object detection with short connections. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 3203–3212 (2017)
18. Howard, A.G., et al.: MobileNets: efficient convolutional neural networks for mobile vision applications. arXiv preprint [arXiv:1704.04861](https://arxiv.org/abs/1704.04861) (2017)
19. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708 (2017)
20. Jia, X., De Brabandere, B., Tuytelaars, T., Gool, L.V.: Dynamic filter networks. In: Conference and Workshop on Neural Information Processing Systems, pp. 667–675 (2016)
21. Ju, R., Liu, Y., Ren, T., Ge, L., Wu, G.: Depth-aware salient object detection using anisotropic center-surround difference. *Signal Process. Image Commun.* **38**, 115–126 (2015)
22. Wei, J., Wang, S., Huang, Q.: F3Net: fusion, feedback and focus for salient object detection. In: AAAI Conference on Artificial Intelligence (2020)
23. Li, G., Yu, Y.: Visual saliency based on multiscale deep features. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 5455–5463 (2015)
24. Li, N., et al.: Saliency detection on light field. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 2806–2813 (2014)
25. Li, Y., Hou, X., Koch, C., Rehg, J.M., Yuille, A.L.: The secrets of salient object segmentation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 280–287 (2014)
26. Liu, J.J., Hou, Q., Cheng, M.M., Feng, J., Jiang, J.: A simple pooling-based design for real-time salient object detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2019)
27. Liu, N., Han, J., Yang, M.H.: PicaNet: learning pixel-wise contextual attention for saliency detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 3089–3098 (2018)
28. Liu, W., Rabinovich, A., Berg, A.C.: ParseNet: looking wider to see better. arXiv preprint [arXiv:1506.04579](https://arxiv.org/abs/1506.04579) (2015)
29. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440 (2015)
30. Mahadevan, V., Vasconcelos, N.: Saliency-based discriminant tracking. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2009)



31. Margolin, R., Zelnik-Manor, L., Tal, A.: How to evaluate foreground maps? In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2014)
32. Niu, Y., Geng, Y., Li, X., Liu, F.: Leveraging stereopsis for saliency analysis. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 454–461 (2012)
33. Pang, Y., Zhao, X., Zhang, L., Lu, H.: Multi-scale interactive network for salient object detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2020)
34. Peng, H., Li, B., Xiong, W., Hu, W., Ji, R.: RGBD salient object detection: a benchmark and algorithms. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8691, pp. 92–109. Springer, Cham (2014). [https://doi.org/10.1007/978-3-319-10578-9\\_7](https://doi.org/10.1007/978-3-319-10578-9_7)
35. Perazzi, F., Krähenbühl, P., Pritch, Y., Hornung, A.: Saliency filters: contrast based filtering for salient region detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 733–740 (2012)
36. Piao, Y., Ji, W., Li, J., Zhang, M., Lu, H.: Depth-induced multi-scale recurrent attention network for saliency detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 7254–7263 (2019)
37. Qu, L., He, S., Zhang, J., Tian, J., Tang, Y., Yang, Q.: RGBD salient object detection via deep fusion. *IEEE Trans. Image Process.* **26**(5), 2274–2285 (2017)
38. Ren, Z., Gao, S., Chia, L.T., Tsang, I.W.H.: Region-based saliency detection and its application in object recognition. *IEEE Trans. Circuits Syst. Video Technol.* **24**(5), 769–779 (2013)
39. Rui, Z., Ouyang, W., Wang, X.: Unsupervised salience learning for person re-identification. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2013)
40. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556) (2014)
41. Wang, L., et al.: Learning to detect salient objects with image-level supervision. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 136–145 (2017)
42. Wang, N., Gong, X.: Adaptive fusion for RGB-D salient object detection. *IEEE Access* **7**, 55277–55284 (2019)
43. Wang, T., Piao, Y., Li, X., Zhang, L., Lu, H.: Deep learning for light field saliency detection. In: Proceedings of the IEEE International Conference on Computer Vision, October 2019
44. Wei, Y., et al.: STC: a simple to complex framework for weakly-supervised semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(11), 2314–2320 (2016)
45. Wu, Z., Su, L., Huang, Q.: Cascaded partial decoder for fast and accurate salient object detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 3907–3916 (2019)
46. Yan, Q., Xu, L., Shi, J., Jia, J.: Hierarchical saliency detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 1155–1162 (2013)
47. Yang, C., Zhang, L., Lu, H., Ruan, X., Yang, M.H.: Saliency detection via graph-based manifold ranking. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 3166–3173 (2013)
48. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. In: Bengio, Y., LeCun, Y. (eds.) International Conference on Learning Representations (2016). <https://arxiv.org/abs/1511.07122>

49. Zhao, J.X., Cao, Y., Fan, D.P., Cheng, M.M., Li, X.Y., Zhang, L.: Contrast prior and fluid pyramid integration for RGBD salient object detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 3927–3936 (2019)
50. Zhao, J.X., Liu, J.J., Fan, D.P., Cao, Y., Yang, J., Cheng, M.M.: EGNNet: edge guidance network for salient object detection. In: Proceedings of the IEEE International Conference on Computer Vision, October 2019
51. Zhao, X., Pang, Y., Zhang, L., Lu, H., Zhang, L.: Suppress and balance: A simple gated network for salient object detection. In: Proceedings of European Conference on Computer Vision (2020)
52. Zhu, C., Li, G.: A three-pathway psychobiological framework of salient object detection using stereoscopic technology. In: International Conference on Computer Vision Workshops, pp. 3008–3014 (2017)
53. Zhu, C., Li, G., Wang, W., Wang, R.: An innovative salient object detection using center-dark channel prior. In: International Conference on Computer Vision Workshops, pp. 1509–1515 (2017)