Rethinking Feature Aggregation for Deep RGB-D Salient Object Detection

Yuan-fang Zhang[a,b], Jiangbin Zheng[a,**], Long Li[c], Nian Liu[d], Wenjing Jia[b], Xiaochen Fan[b], Chengpei Xu[b], Xiangjian He[b,**]

[a]*School of Computer Science, Northwestern Polytechnical University, P.R.China*
[b]*Faculty of Engineering and IT, University of Technology Sydney, Australia*
[c]*School of Automation, Northwestern Polytechnical University, P.R.China*
[d]*Mohamed bin Zayed University of Artificial Intelligence, UAE*

Abstract

Two-stream UNet based architectures are widely used in deep RGB-D salient object detection (SOD) models. However, UNet only adopts a top-down decoder network to progressively aggregate high-level features with low-level ones. In this paper, we propose to enrich feature aggregation via holistic aggregation paths and an extra bottom-up decoder network. The former aggregates multi-level features holistically to learn abundant feature interactions while the latter aggregates improved low-level features with high-level features, thus promoting their representation ability. Aiming at the two-stream architecture, we propose another early aggregation scheme to aggregate and propagate multi-modal encoder features at each level, thereby improving the encoder capability. We also propose a factorized attention module to efficiently modulate the feature aggregation action for each feature node with multiple learned attention factors. Experimental results demonstrate that all of the proposed components can gradually improve RGB-D SOD results. Consequently, our final SOD model performs favorably against other state-of-the-art methods.

Keywords: RGB-D saliency detection, UNet, Feature aggregation, Gated attention.

1. Introduction

Salient object detection focuses on localizing and segmenting the most distinctive object(s) in a visual scene. It mimics the human visual attention mechanism to efficiently allocate visual processing resources on informative visual elements. Thus, SOD can be used as a preprocessing technique and supply informative cues for many other computer vision tasks, such as object detection [1], video object segmentation [2], semantic segmentation [3, 4], image editing [5] and intelligent vision surveillance in smart city application[6].

Most SOD models [7, 8, 9, 10, 11, 12, 13] typically detect salient objects from RGB images. In a pioneer work of [14], Ouerhani and Hugli showed that depth could also supply useful cues and largely boost the performance for saliency detection. This is also intuitive since human beings live in a real 3D environment and depth largely impacts our perception of visual scenes. Many subsequent saliency models, *e.g.*, those in [15, 16, 17, 18], have started to leverage RGB-D images for saliency detection. Recently, Convolutional Neural Networks (CNNs) have widely been seen in the computer vision community and have also shown excellent performance on various computer vision tasks. Hence, many works have also introduced two-stream CNNs for RGB-D SOD to exploit their powerful feature learning capability.

Some deep models [21, 22] applied the two-stream Fully Convolutional Network (FCN) [19] architecture to feedforward each input RGB-D image pair into two CNN streams and directly obtained the saliency map by fusing their final feature maps, as shown in Figure 1(a). FCN processes the input image pair in a bottom-up manner, progressively extracting low-level features in shallow layers and high-level features in deep layers. Although it is simple and straightforward, the single path of the bottom-up information flow heavily limits the model performance since usually the final feature map of a CNN is very coarse, thus the obtained saliency map lacks object details.

Considering the multi-level feature maps spontaneously obtained by each CNN, most of other works [23, 24, 25, 26, 27] have adopted the two-stream UNet [20] architecture to aggregate multi-level features for RGB-D SOD. As shown in Figure 1(b), the two-stream UNet first uses two encoder networks to extract multi-level image features in a bottom-up manner. Then, there is one or two decoder networks to successively aggregate high-level features with low-level ones in a top-down processing and simultaneously

*Corresponding author
**Corresponding author
Email addresses: zyf.robinzhang@gmail.com (Yuan-fang Zhang), zhengjb@nwpu.edu.cn (Jiangbin Zheng), longli.nwpu@gmail.com (Long Li), liunian228@gmail.com (Nian Liu), Wenjing.Jia@uts.edu.au (Wenjing Jia), fanxiaochen33@gmail.com (Xiaochen Fan), Chengpei.Xu@student.uts.edu.au (Chengpei Xu), Xiangjian.He@uts.edu.au (Xiangjian He)

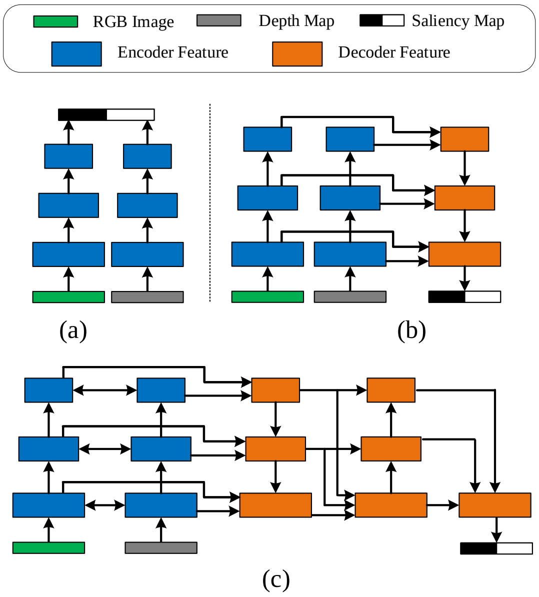


Figure 1: **Comparison of different network architectures.** (a) Two-stream FCN [19]. (b) Two-stream UNet [20]. (c) Our proposed network. We cascade both top-down and bottom-up feature aggregation for deep RGB-D SOD to further leverage improved low-level features for promoting high-level features. We also propose to holistically aggregate features across all levels to learn plentiful multi-level feature interactions. Early aggregation paths are also presented to aggregate and propagate cross-modal encoder features.

fuse cross-modal features. In each decoder module, the features of its symmetric encoder module at the same level are reused through a skip connection and fused with previous decoder features. As such, discriminative semantic information in deep layers can be effectively integrated with local structures in shallow layers through the top-down propagation, thus enabling both accurate object localization and precise shape and boundary segmentation.

However, UNet carries out top-down feature aggregation only once. Only high-level information can be aggregated with low-level features to improve their representation ability in the decoder, while the high-level features themselves cannot be improved. To solve this problem, in this paper, we propose to add an additional bottom-up aggregation path, in which the improved low-level features from the top-down path are propagated again to high-level layers, as shown in Figure 1(c). As we cascade both bottom-up and top-down feature aggregation, the features across all levels can be gradually improved.

Another problem is that above networks only gradually aggregate features at every two adjacent levels. Although this feature aggregation scheme avoids large scale changes and is widely used in previous works, we argue that it limits direct feature interactions among multi-level features. To alleviate this issue, we further propose holistic aggregation paths to holistically aggregate multi-level features after the bottom-up and top-down processing. Thus, the network can learn abundant cross-level feature fusion mechanism for SOD by considering them all at the same time.

Considering the two-stream architecture, the authors of existing works usually simply adopt two-stream encoders independently and only conduct feature aggregation in the decoding phase [21, 27, 28]. Or they fuse cross-modal encoder features to reuse them in decoders [29, 30, 31], without improving other encoder features. This is because they use pretrained CNN models as encoders and they require to preserve their network structures and pretrained parameters. In this paper, we aggregate and propagate cross-modal features at the early stage, i.e., in the encoding phase. We adopt a residual-learning based aggregation scheme to aggregate cross-modal encoder features and propagate them back to the original encoder paths, hence enhancing the feature capability from the very beginning.

Furthermore, previous work usually aggregate features by directly concatenating [23, 24, 25] or adding [32] them together. However, not all aggregated features are helpful for the final SOD task. We propose to generate gated attention for all of the involved features to modulate the aggregation flow at every node. To reduce the amount of the required gated attention weights and the computation and memory costs, we propose to factorize the gate matrix into the multiplication of channel-wise and spatial gates with multiple factors. This proposed multi-factored gated attention mechanism learns different gates in different factors and thus can ensemble multiple attention models to make a better decision.

At last, we summarize the main contributions of this work as follows.

- We propose a novel feature aggregation architecture for RGB-D SOD. We cascade both bottom-up and top-down feature aggregation paths and also introduce holistic aggregation paths, which promote both low-level and high-level features and boost multi-level feature interactions. An early aggregation scheme is also presented to enhance the two-stream encoders.
- We propose a novel factorized gated attention model for modulating the feature aggregation actions. We factorize the gated attention weight matrix of each feature map as the multiplication of two multi-factored channel-wise and spatial gate matrices. As such, both computational costs and model effectiveness are improved.
- We conduct experiments on eight widely used RGB-D SOD benchmark datasets. Experimental results demonstrate that all of the proposed model components can gradually improve the model performance. Consequently, our final model outperforms other state-of-the-art methods.

In the subsequent sections, in Section II we first discuss our model with related work. Then, we present our model in Section III and report the experimental results in Section IV. Finally, in Section V we draw our conclusion.

2. Related Work

CNNs have been widely used for RGB SOD and RGB-D SOD. For the former, please refer to [33] for a comprehensive survey. We focus on the latter in this paper. In two early pioneering deep RGB-D SOD works [34, 35], the authors used superpixels as the computational units and combined both traditional handcrafted features and CNNs to classify them as salient or non-salient. However, such schemes are usually computationally inefficient and therefore limit the model performance. Subsequent models start to adopt CNNs to directly process each input image and obtain the saliency map. Specifically, Han *et al.* [21] adopted two-stream CNNs to process RGB and depth images respectively, and then used fully connected layers to predict global saliency maps. Chen *et al.* [36] further combined this method with FCNs to fuse global and local contextual reasoning. In [22], Fan *et al.* first depurated depth maps and then use single-stream FCNs with Pyramid Dilated Convolution modules [37] to predict saliency maps. These models directly predict saliency maps from the last layer of a CNN without considering multi-level features.

Most of the other works use the UNet architecture to gradually aggregate multi-level deep features. For instance, Chen *et al.* [24] first used two encoder networks to extract multi-level features from an RGB image and a depth image, respectively. Then, they proposed to densely fuse multi-level cross-modal features in a top-down decoder network. Zhao *et al.* [38] first proposed to leverage depth-based contrast to enhance the RGB encoder features, and then fused multi-level features using a top-down decoder with dense short connections. In [25], Liu *et al.* followed the work in [9] to embed recurrent convolutional layers into top-down decoder modules for fusing encoder and decoder features with the depth map. Li *et al.* [31] fused RGB and depth encoder features first and then also adopted a UNet style decoder to aggregate the multi-level features. All of these models only considered a top-down feature aggregation path for RGB-D SOD, without exploring other feature aggregation schemes. In contrast, we cascade both top-down and bottom-up processings to promote features at all levels. Furthermore, most previous works directly use pretrained two-stream encoder networks without both fusing and improving encoder features, except for [36]. However, the authors of [36] only propagated depth encoder features to RGB ones, while we perform bidirectional feature aggregation and propagation via the proposed early aggregation scheme.

Attention models are also widely used in RGB-D SOD models. Chen *et al.* [23] adopted SENet [39] style channel attention in decoder modules to modulate feature channels. In [32], channel attention and spatial attention were separately adopted in a recurrent attention module for generating the final saliency maps. Liu *et al.* [40] proposed to selectively fuse self-mutual attention for fusing cross-modal information at the beginning of the decoder network. Different from the existing models, we propose to modulate the whole feature map in each decoder module with gated attention and further present a multi-factored factorization mechanism to save computational costs and enhance the model capability.

In [41, 42, 43], gated attention were also used in the convolution operation for language modeling, image inpainting, and RGB-D SOD, respectively. Different from them, we propose the multi-factored factorization operation for gated attention to reduce computational costs and boost the model capability.

Two works are closely related to our proposed model. Chen and Li [44] also used both top-down and bottom-up decoders. The difference between our model and theirs are as follows. First, they adopted the bottom-up decoder first to fuse cross-modal features and then used the top-down decoder to obtain coarse-to-fine saliency maps, while we build our model based on UNet and use the top-down decoder first. Second, we also propose to use the holistic aggregation paths to aggregate all-level features simultaneously, while they only linearly fused the side output saliency maps. Third, they used the existing SENet [39] style channel attention in the top-down decoder while we propose a novel factorized gated attention model and employ it in all aggregation paths. Forth, we also propose an early aggregation scheme to promote the two-stream encoders. Another work is that in [45], Wang *et al.* proposed to iterate top-down and bottom-up decoders for multiple steps for RGB SOD. Different from them, we only cascade top-down and bottom-up decoding paths once and found the model performance already saturated. Furthermore, they adopt RNN in each decoder module to enhance the decoder capability while we use the proposed gated attention mechanism. We also propose the holistic aggregation paths to more effectively leverage multi-level features and present the early aggregation scheme for the two-stream architecture nature of the RGB-D SOD models.

3. Proposed Method

In this section, we articulate the proposed network for RGB-D SOD. Its detailed network architecture is shown in Figure 2.

3.1. Encoder Network

We first follow most previous methods and adopt a two-stream encoder network for extracting multi-level RGB and depth features. In order to learn common features for cross-modality, we share the network structure and parameters for the two encoder branches. To leverage better image features, we use an ImageNet [46] pretrained network as the encoder. The VGG 16-layer network [47] is adopted for a fair comparison with previous works. It has five convolutional (Conv) blocks and pooling layers, and two fully connected (FC) layers. For better adapting the network to SOD, we enhance the original VGG network by keeping large scale feature maps and preserving high-level FC layers. Concretely, we first reduce the stride of the pool5 layer to 1. Then, we convert the FC6 layer to a

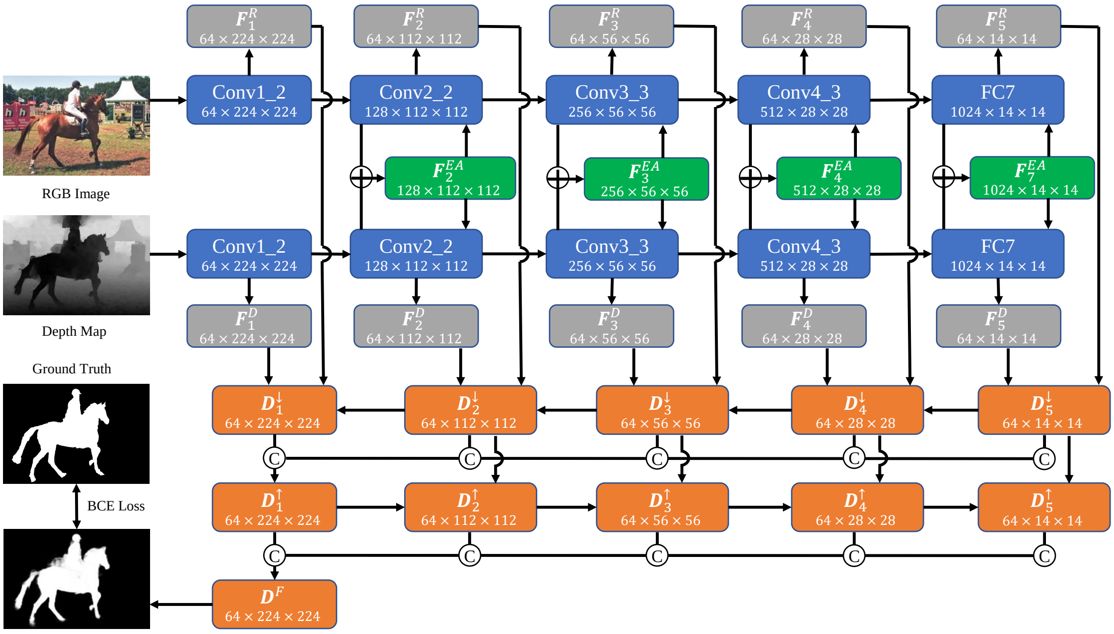


Figure 2: **Network architecture of the proposed RGB-D SOD model.** We first use two encoder branches for the RGB and depth inputs to extract multi-level encoder features ($\boldsymbol{F}_*^R$ and $\boldsymbol{F}_*^D$). Within the two-stream encoders, we adopt early aggregation paths ($\boldsymbol{F}_*^{EA}$) to propagate cross-model information from the very beginning. Here, the early aggregation path for the two Conv5_3 layers is not shown. Then, we successively adopt a top-down decoder network ($\boldsymbol{D}_*^{\downarrow}$) and a bottom-up one ($\boldsymbol{D}_*^{\uparrow}$) to aggregate multi-level features. We also use holistic aggregation paths to directly aggregate features across all levels. The size of each feature map is also given and denoted by *channel* × *height* × *width*. Ⓒ denotes concatenation and ⊕ means element-wise summation.

Conv layer with 1024 channels and 3×3 kernels, and adopt the dilated convolution algorithm [48] with $dilation = 6$. Similarly, the FC7 layer is also converted to a Conv layer with 1024 channels and 1×1 kernels. As such, the stride of the encoder network is reduced from 32 to 16 and high-level FC features are also preserved in the encoder.

To propagate cross-modal information from an early stage, we introduce early aggregation (EA) into the two encoders, specifically for the last Conv feature maps of the last four Conv blocks and the FC7 layer, which are Conv2_2, Conv3_3, Conv4_3, Conv5_3, and FC7 layers. We do not use EA for the first Conv block since its low-level features may be quite different in the two modalities while the other higher layers can learn more common semantics. Given an RGB encoder feature map and a depth one from the same level, which are named as $\boldsymbol{E}_i^R$ and $\boldsymbol{E}_i^D$, respectively, our EA path first aggregates them by element-wise summation and averaging, obtaining the EA feature map:

$$\boldsymbol{F}_i^{EA} = \frac{\boldsymbol{E}_i^R + \boldsymbol{E}_i^D}{2}. \tag{1}$$

Then, we propagate $\boldsymbol{F}_i^{EA}$ back to the two encoder features using residual learning:

$$\begin{aligned} \boldsymbol{E}_i^R &= \boldsymbol{E}_i^R + \alpha \cdot Conv(\boldsymbol{F}_i^{EA}), \\ \boldsymbol{E}_i^D &= \boldsymbol{E}_i^D + \alpha \cdot Conv(\boldsymbol{F}_i^{EA}). \end{aligned} \tag{2}$$

Here, the two $Conv$ means two 1×1 Conv layers and α is a learnable parameter. We initialize α to 0 to make sure that the EA path brings no impact to the pretrained encoder networks at the beginning of the model training. As such, the EA path boosts the encoder representation ability by leveraging cross-modal information and leveraging the pretrained model parameters losslessly.

Finally, we pick out the output feature maps of the Conv1_2, Conv2_2, Conv3_3, Conv4_3, and FC7 layers as the multi-level features and reuse them in later decoders. Since these features have diverse channel numbers, we first use 3×3 Conv layers to convert each of them to 64 channels, thus making them compatible with each other in the subsequent feature aggregation. For representation simplicity, we denote these multi-level features by $\boldsymbol{F}_1^R$ to $\boldsymbol{F}_5^R$ and $\boldsymbol{F}_1^D$ to $\boldsymbol{F}_5^D$ for the RGB and the depth branches, respectively, as shown in Figure 2. The input scales of each RGB image and the depth map are fixed to 224×224 for simplicity. Hence, the sizes of the multi-level feature maps can be easily inferred, as marked in Figure 2.

3.2. Decoder Networks

After obtaining the ten multi-level features from both of the RGB and the depth branches, we aggregate them for RGB-D SOD. First, we follow UNet [20] to progressively aggregate features at every two adjacent levels in a top-down (denoted as ↓) decoder network. Specifically, in the i^{th} top-down decoder module, where $i \in \{1, 2, 3, 4\}$, we obtain its decoder feature $\boldsymbol{D}_i^{\downarrow}$ by aggregating the previous decoder feature $\boldsymbol{D}_{i+1}^{\downarrow}$ with the RGB and depth features $\boldsymbol{F}_i^R$ and $\boldsymbol{F}_i^D$ at this level. Since $\boldsymbol{D}_{i+1}^{\downarrow}$ has a smaller spatial size, we first upsample it by bilinear interpolation. For the 5^{th} decoder module, we directly aggregate $\boldsymbol{F}_5^R$ and $\boldsymbol{F}_5^D$. The top-down feature aggregation process can be summarized by equation(3):

$$\boldsymbol{D}_i^{\downarrow} = \begin{cases} Conv(BR([\boldsymbol{F}_i^R, \boldsymbol{F}_i^D])), & i = 5, \\ Conv(BR([UP(\boldsymbol{D}_{i+1}^{\downarrow}), \boldsymbol{F}_i^R, \boldsymbol{F}_i^D])), & i \in \{1,2,3,4\}, \end{cases} \tag{3}$$

where $[,]$ means the concatenation operation, BR means batch normalization [49] and ReLU, $Conv$ denotes a 3×3 Conv layer with 64 channels and UP means bilinear upsampling.

After the top-down feature aggregation, low-level features can be enhanced by high-level features. Thus, the final output feature map $\boldsymbol{D}_1^{\downarrow}$ simultaneously preserves local details and contains high-level semantics. Most of previous works directly use this layer to predict the saliency maps. We further construct a bottom-up (denoted as $\uparrow$) decoder network to use the enhanced low-level features to improve the high-level features. To be concrete, we first use holistic aggregation paths to aggregate the features $\boldsymbol{D}_i^{\downarrow}$ at all levels to obtain the first feature map $\boldsymbol{D}_1^{\uparrow}$. Then, in the subsequent $i \in \{2,3,4,5\}$ bottom-up decoder modules, we generate the decoder features $\boldsymbol{D}_i^{\uparrow}$ by aggregating the previous bottom-up decoder feature $\boldsymbol{D}_{i-1}^{\uparrow}$ with the top-down decoder feature $\boldsymbol{D}_i^{\downarrow}$ at this level. Since $\boldsymbol{D}_{i-1}^{\uparrow}$ has a larger spatial size, we downsample it using a max-pooling layer with stride of 2. The bottom-up feature aggregation process can be represented by equation(4):

$$\boldsymbol{D}_i^{\uparrow} = \begin{cases} Conv(BR([\boldsymbol{D}_1^{\downarrow}, UP(\boldsymbol{D}_2^{\downarrow}), \cdots, UP(\boldsymbol{D}_5^{\downarrow})])), & i = 1, \\ Conv(BR([DW(\boldsymbol{D}_{i-1}^{\uparrow}), \boldsymbol{D}_i^{\downarrow}]), & i \in \{2,3,4,5\}, \end{cases} \tag{4}$$

where DW means down-sampling with a max-pooling layer.

After the bottom-up feature aggregation, high-level features can also perceive better low-level features thus generating better semantic information. Hence, by cascading both of the top-down and bottom-up decoder networks, we can simultaneously enhance all low-level and high-level features. Finally, we adopt the holistic aggregation again at the finest scale to obtain the final decoder feature map as equation(5):

$$\boldsymbol{D}^F = Conv(BR([\boldsymbol{D}_1^{\uparrow}, UP(\boldsymbol{D}_2^{\uparrow}), \cdots, UP(\boldsymbol{D}_5^{\uparrow})])). \tag{5}$$

A 1×1 Conv layer with 1 channel and the Sigmoid activation function can be used on top of $\boldsymbol{D}^F$ to obtain the final saliency map. During training, we also generate an intermediate saliency map from $\boldsymbol{D}_1^{\uparrow}$ in the same way. Then, we compute two binary cross entropy losses between the two saliency maps and the ground truth to train the whole network.

3.3. Factorized Gated Attention

It is worth noting that SOD is a challenging dense prediction task, and usually not all features are useful for the final decision. Thus, we propose to introduce gated attention for the feature aggregation operations to adaptively select informative features for each decoder module. Specifically, for the $Conv$ layers in (2), (3), (4), (5), considering an input feature map $\boldsymbol{X} \in \mathbb{R}^{C \times H \times W}$, in which C, H, and W respectively denote its channel number, height, and width, we predict an gated attention matrix $\boldsymbol{G}$ of the same size with each of its element in the range of $[0, 1]$. Then, we use $\boldsymbol{G}$ to modulate each node of $\boldsymbol{X}$ to control the aggregation flow in each decoder module as equation(6):

$$\boldsymbol{X}^G = \boldsymbol{G} \odot \boldsymbol{X}, \tag{6}$$

where $\odot$ is the element-wise multiplication. As such, $\boldsymbol{G}$ serves as a modulator and can retain informative features and suppress useless ones in $\boldsymbol{X}$. Then, we use $\boldsymbol{X}^G$ as the input for the $Conv$ layers.

However, predicting $\boldsymbol{G}$ requires predicting all of the $C \times H \times W$ gate weights. A straightforward way is using a Conv layer with C channels on $\boldsymbol{X}$. Nevertheless, this scheme only uses local information, which equals to generating channel-wise gates for each pixel with shared parameters. Another way is to use an FC layer. This design is computationally prohibitive since it requires a large number of parameters to learn. We propose to learn a factorized form of $\boldsymbol{G}$ for reducing the number of attention weights to predict. Concretely, we factorize $\boldsymbol{G} \in \mathbb{R}^{C \times H \times W}$ into the multiplication of two low-rank matrices $\boldsymbol{G}^c \in \mathbb{R}^{C \times r}$ and $\boldsymbol{G}^s \in \mathbb{R}^{r \times (H \times W)}$. In this way, when a small number is used for r, the number of gate weights to predict can be reduced to $(C + H \times W) \times r$. For example, for $\boldsymbol{D}_2^{\downarrow}$ where $C = 192, W = H = 112$, using our factorization scheme with 2 factors, we can decrease the computational costs by 94.6 times.

Using the factorized attention, equation (6) can be rewritten to:

$$\begin{aligned} \boldsymbol{X}^G &= \boldsymbol{G} \odot \boldsymbol{X} \\ &= (\boldsymbol{G}^c \boldsymbol{G}^s) \odot \boldsymbol{X} \\ &= \sum_{j=1}^{r} (\boldsymbol{G}_j^c (\boldsymbol{G}_j^s)^\top) \odot \boldsymbol{X}, \end{aligned} \tag{7}$$

where $\boldsymbol{G}_j^c \in \mathbb{R}^C$ and $\boldsymbol{G}_j^s \in \mathbb{R}^{(H \times W)}$ are the j^{th} factors of $\boldsymbol{G}^c$ and $\boldsymbol{G}^s$, respectively. We can respectively regard $\boldsymbol{G}_j^c$ and $\boldsymbol{G}_j^s$ as the traditional channel and spatial gated attention. In this way, $\boldsymbol{G}$ can be seen as being spanned by the outer product of channel attention and spatial attention. As such, we efficiently generate the attention weights for the entire feature map and leads to cheaper computation and memory costs. Furthermore, we generate r factors for both channel and spatial gated attention, which is similar to the multi-head attention in [50]. Thus, our proposed factorized gated attention (FGA) mechanism can help to select different channels and spatial locations in different factors.

Motivated by the SENet model [39], we use average pooling and an FC layer to predict $\boldsymbol{G}^c$. Specifically, we

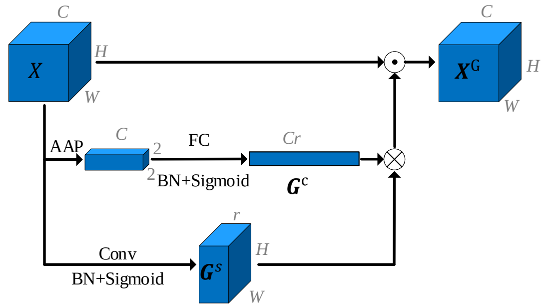


Figure 3: **Architecture of the proposed factorized gated attention module.** We factorize the gated attention of the feature map $\boldsymbol{X}$ as the multiplication of multi-factored channel-wise gate weights $\boldsymbol{G}^c$ and spatial gate weights $\boldsymbol{G}^s$ to reduce computation and memory costs and introduce attention ensemble. AAP: adaptive average pooling. $\odot$: element-wise multiplication. $\otimes$: matrix multiplication. Sizes of some crucial features are marked by gray font.

first adopt adaptive average pooling on $\boldsymbol{X}$ to pool the entire feature map to the spatial size of 2×2. The resultant feature map represents the mean activation value of each channel in a $\frac{H}{2} \times \frac{W}{2}$ window. Then, we use an FC layer with BN and the Sigmoid activation function to generate $\boldsymbol{G}^c$, which is a vector of $C \times r$ dimensions. For generating $\boldsymbol{G}^s$, we first use a 7×7 Conv layer with r channels on $\boldsymbol{X}$. Then, BN and the Sigmoid activation function are used to obtain $\boldsymbol{G}^s$. Figure 3 shows the detailed architecture of the proposed FGA module.

Since each element of $\boldsymbol{G}^c$ and $\boldsymbol{G}^s$ is in the range of $[0, 1]$ and the summation over r factors in (7) will magnify the value range of the elements of $\boldsymbol{G}$, we further divide $\boldsymbol{G}$ by r to shrink its value range back to $[0, 1]$. The final formulation of the proposed FGA module in equation(8):

$$\boldsymbol{X}^G = \frac{1}{r}(\boldsymbol{G}^c \boldsymbol{G}^s) \odot \boldsymbol{X}. \tag{8}$$

We write a new layer for this operation to implement it efficiently. Given $\partial L / \partial \boldsymbol{X}^G$ be the gradient of the loss function L with respect to $\boldsymbol{X}^G$, the gradients with respect to the three inputs can be easily obtained by the chain rule as equation(9):

$$\begin{aligned} \frac{\partial L}{\partial \boldsymbol{X}} &= \frac{1}{r}(\boldsymbol{G}^c \boldsymbol{G}^s) \odot \frac{\partial L}{\partial \boldsymbol{X}^G}, \\ \frac{\partial L}{\partial \boldsymbol{G}^c} &= \frac{1}{r}(\frac{\partial L}{\partial \boldsymbol{X}^G} \odot \boldsymbol{X})(\boldsymbol{G}^s)^\top, \\ \frac{\partial L}{\partial \boldsymbol{G}^s} &= \frac{1}{r}(\boldsymbol{G}^c)^\top(\frac{\partial L}{\partial \boldsymbol{X}^G} \odot \boldsymbol{X}). \end{aligned} \tag{9}$$

Thus, the proposed FGA module can be trained along with other layers of the network simultaneously via existing gradient based optimizers.

We adopt FGA for all decoder modules and the generation of the multi-level encoder features $\boldsymbol{F}^R_*$ and $\boldsymbol{F}^D_*$. Experimental results in Section 4.4 demonstrate that it can further improve the feature aggregation effectiveness for RGB-D SOD.

4. Experiments

4.1. Datasets

We evaluate the effectiveness of the proposed model on eight widely used RGB-D SOD benchmark datasets. The first one is the **NJUD** [51] dataset, which has 1985 stereo images. The images are selected from the Internet, 3D movies, and stereo photographs. The salient objects are labeled in a 3D display environment. The second one is the **NLPR** [52] dataset with 1000 RGB-D images collected by Microsoft Kinect. Most of them are indoor images with simple salient objects. The third one is the **RGBD135** [17] dataset, which has 135 RGB-D indoor images captured by Kinect. The fourth one is the **LFSD** [53] dataset. It consists of 100 challenging images captured by the Lytro light field camera, including 60 indoor scenes and 40 outdoor scenes. The fifth one is the **STERE** [54] dataset, which has 1000 stereoscopic images. Many of the images include complex scenes and various objects. **SSD** [55] is the sixth dataset that has 80 images selected from three stereo movies. **DUT-RGBD** [32] dataset is the seventh one. It includes 800 indoor and 400 outdoor images with challenging scenes and generated depth maps. The last one is **SIP** [56] dataset, which is a newly released one with 1000 human activities oriented images.

4.2. Implementation Details

We follow the previous work [32] to select 1400, 650, and 800 images from the **NJUD**, **NLPR**, and **DUT-RGBD** datasets, respectively, to train the proposed SOD network. To alleviate overfitting, we conduct data augmentation by first resizing each training image pair to 288×288 pixels and then randomly cropping 224×224 image patches and also use random horizontal flipping. The input image pairs are pre-processed by subtracting the mean RGB and depth pixels computed on the training set. We adopt the stochastic gradient descent (SGD) algorithm with momentum to train our network, where we set the batchsize, momentum, and weight decay to 4, 0.9, and 0.0005, respectively. We set the initial learning rate of the VGG part of the two encoder branches as 0.001 and train the other part of the network with random initialization and the initial learning rate of 0.01. We train the network with totally 60,000 steps and reduce the learning rates by 10 times at the $40,000$th and $50,000$th steps, respectively.

Our code is implemented based on an improved Caffe [57] library[1] to save GPU memory. We use a GTX 1080 Ti GPU to accelerate network training and testing. During testing, we directly resize each image pair to 224×224 pixels as the input and get the network output as the predicted saliency map, without any post-processing technique. The testing process costs 0.089 seconds for each image.

[1] https://github.com/yjxiong/caffe

Table 1: Ablation study on the effectiveness of the holistic aggregation paths (HA), the bottom-up aggregation (BU), the factorized gated attention (FGA), and the early aggregation (EA). Blue indicates the best performance.

ID	Settings				NJUD [51]				NLPR [52]				DUT-RGBD [32]				STERE [54]			
	HA	BU	FGA	EA	S_m	maxF	E_ξ	MAE	S_m	maxF	E_ξ	MAE	S_m	maxF	E_ξ	MAE	S_m	maxF	E_ξ	MAE
I					0.888	0.889	0.930	0.059	0.908	0.894	0.951	0.036	0.898	0.906	0.937	0.052	0.891	0.888	0.936	0.055
II	✓				0.894	0.892	0.933	0.053	0.911	0.902	0.953	0.035	0.912	0.915	0.948	0.046	0.889	0.890	0.937	0.055
III	✓	✓			0.897	0.890	0.929	0.051	0.917	0.901	0.950	0.030	0.915	0.914	0.944	0.041	0.897	0.887	0.932	0.048
IV	✓	✓	$r=1$		0.899	0.890	0.928	0.048	0.914	0.894	0.944	0.031	0.918	0.921	0.949	0.042	0.897	0.887	0.934	0.049
V	✓	✓	$r=2$		0.901	0.893	0.933	0.047	0.920	0.901	0.953	0.029	0.921	0.926	0.952	0.037	0.905	0.897	0.941	0.043
VI	✓	✓	$r=3$		0.903	0.894	0.934	0.047	0.919	0.903	0.953	0.029	0.919	0.919	0.946	0.040	0.902	0.892	0.938	0.046
VII	✓	✓	$r=2$	✓	0.906	0.902	0.936	0.045	0.927	0.912	0.961	0.025	0.926	0.927	0.954	0.034	0.904	0.896	0.940	0.042

4.3. Evaluation Metrics

We adopt four widely used SOD metrics. The first one is the max F-measure score. Concretely, for each image, we first use a series of thresholds, which vary from 0 to 1 to binarize the predicted saliency map. Then, we compare the binarized saliency maps with the ground truth saliency map, thus obtaining a series of precision-recall value pairs. F-measure comprehensively considers both precision and recall as equation(10):

$$F_\beta = \frac{(1+\beta^2)Precision \times Recall}{\beta^2 Precision + Recall}, \tag{10}$$

where β^2 is set to 0.3 as suggested in previous work to emphasize more on precision. Max F-measure F_β^{max} is obtained by selecting the highest F-measure score under the optimal threshold.

The second metric is the Mean Absolute Error (MAE), which computes the average absolute difference between the predicted saliency map $\boldsymbol{S}$ and the ground truth saliency map $\boldsymbol{G}$ as equation(11):

$$MAE = \frac{1}{WH}\sum_{w=1}^{W}\sum_{h=1}^{H}\left|\boldsymbol{G}(w,h) - \boldsymbol{S}(w,h)\right|. \tag{11}$$

Although being widely used in previous work, the above two mentioned metrics are all based on pixel-wise errors and ignore structural information, and they are shown to be highly sensitive for the human visual system. Thus, we use the Structure-measure S_m [58] as our third metric to evaluate the structural similarity between the predicted saliency maps and the ground truth maps.

Fan *et al.* [59] recently simultaneously evaluate image-level statistics and local pixel matching with the proposed Enhanced-alignment measure E_ξ, which demonstrated superiority over other existing measures. Thus, we also follow recent work to adopt this measure as the forth metric.

4.4. Component Analysis

In this part, we analyze the effect of each proposed model component on four large datasets to verify their effectiveness. We use the two-stream UNet [20] as the baseline model, as shown in Row(I) of Table 1.

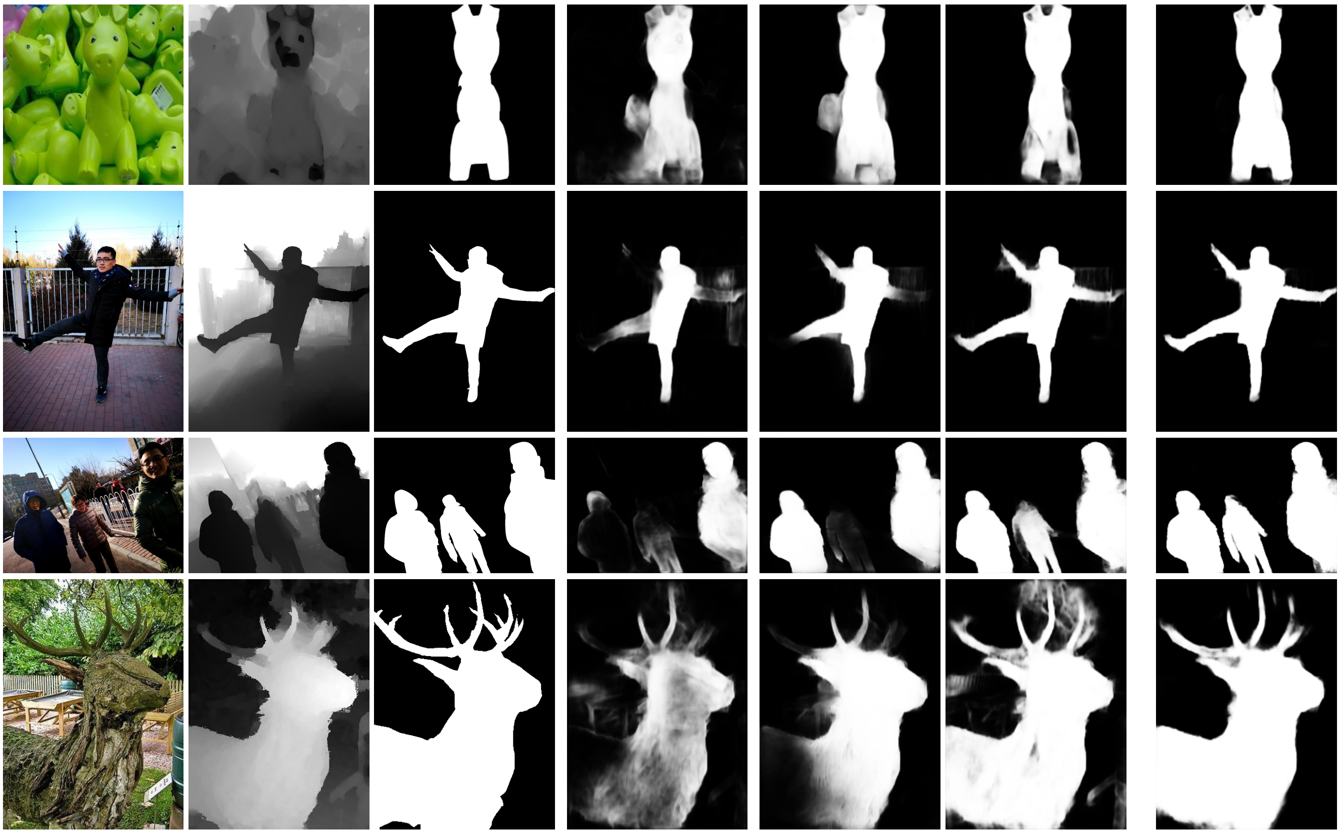
Holistic Aggregation Paths. To evaluate the effectiveness of the proposed holistic aggregation paths, we directly aggregate decoder features across all levels of the UNet model on the finest level and use the obtained feature map (i.e., $\boldsymbol{D}_1^{\uparrow}$) to generate saliency maps. The results are shown in row (II) of Table 1. By comparing them with the results in row (I), we can see that aggregating multi-level features holistically can improve the performance of UNet, especially on the DUT-RGBD [32] dataset.

Bottom-up Aggregation. We further add the bottom-up decoder network to promote high-level features using low-level features from the top-down decoder network of UNet. The results in row (III) show obvious performance gains based on the model setting in row (II), which demonstrates the effectiveness of an additional bottom-up feature aggregation path.

Factorized Gated Attention. We further adopt our proposed factorized gated attention in all decoder modules to verify its effectiveness. We have tried different settings with the factor number r varying from 1 to 3 and show the results in rows (IV) to (VI) of Table 1. We can see that when using 1 factor to factorize the gated attention, the model does not bring obvious performance gains when compared with the results in row (III). However, when we increase the factor number to 2 and 3, the model performance can be obviously improved. We also observe that the model performance saturates when r is greater than 2. Thus, we do not try other settings for r and select $r=2$ as the best setting.

Early Aggregation. The above model settings follow most previous works to use the original VGG network as encoders. Then, we add early aggregation paths between our two-stream encoders to introduce early cross-modal information interaction. The results are given in row (VII) of Table 1. We can see that adding early aggregation paths can effectively improve the model performance on most datasets. Thus, we select this model setting as our final SOD model.

Qualitative Comparison. To further demonstrate the effectiveness of the proposed model components, we show a visual comparison in Figure 4. We can see that adopting



(a)Image (b)Depth (c)GT Saliency (d)Two-stream UNet (e)+HA+BU (f)+HA+BU+FGA (g)+HA+BU+FGA+EA

Figure 4: **Visual comparison of different model settings.** We compare the results of the baseline Two-stream UNet (d), adding the holistic aggregation paths and the bottom-up aggregation (e), and further adding the factorized gated attention (f).

the proposed holistic aggregation, the bottom-up aggregation, the factorized gated attention, and the early aggregation can gradually improve the SOD results. We observe that the proposed model components can help not only recover missing salient regions, but also filter out redundant detected regions. As a result, the final model can obtain better saliency maps that are close to the ground truth.

What do the multi-factored attention learn? Since we factorize the gated attention into the multiplication of channel-wise gated attention $\boldsymbol{G}^c$ and a spatial gated attention $\boldsymbol{G}^s$ with multiple factors, What do these multiple attention factors learn? To answer this question, we show the learned two spatial attention maps of our final SOD model in Figure 5 for the $\boldsymbol{D}_2^{\uparrow}$ feature map. We can see that the spatial attention maps mainly focus to highlight object boundaries. The two attention maps in each example are slightly different. Thus, our proposed multi-factored attention model can be seen as an ensemble of multiple submodules, which has been widely proved to be useful in various machine learning algorithms. We also observe similar phenomena for the spatial attention in other layers and the channel-wise gated attention.

Comparison between FGA and existing attention models. We compare our proposed FGA with conventional convolutional gated attention (CGA), spatial attention (SA), and the Convolutional Block Attention Module (CBAM) [60], in terms of both model performance and computational costs. For CGA, we simply use a 7×7 Conv layer to generate the gated attention weights with the same size with each input feature map. The attention generation for SA is similar, except that we generate a single channel attention map. For CBAM, we use the default settings to incorporate cascaded channel and spatial attention. We substitute FGA in our SOD model with these three attention models and report the comparison results in Table 2. The results clearly show that our proposed FGA model achieves the best RGB-D SOD performance. In terms of computational costs, we can see that FGA uses much less GPU memory than CGA and is much faster than CGA and CBAM. Compared with CGA, FGA predicts much fewer attention weights. Compared with CBAM, FGA only needs to carry out the attending operation once while CBAM needs to do it twice. Compared with SA, FGA costs a little more inference time but achieves better model performance.

4.5. Comparison with State-of-the-art Models

To verify the effectiveness of our final model for RGB-D SOD, we conduct a performance comparison with other 11 state-of-the-art RGB-D SOD methods. We consider

(a1)Image (b1)Depth (c1)Att 1 (d1)Att 2 (a2)Image (b2)Depth (c2)Att 1 (d2)Att 2

Figure 5: **Visualization of two learned two spatial attention factors for $D_2^{\uparrow}$.** "Att 1" and "Att 2" denote the two spatial attention maps, respectively.

Table 2: Comparison between FGA and the existing attention models, including convolutional gated attention (CGA), spatial attention (SA), and the Convolutional Block Attention Module (CBAM). We report both RGB-D SOD performance and computational costs, which include both memory costs and running times during testing. Here, we only test the network forwarding time and ignore the time for reading and writing images for rigorous comparisons. Blue indicates the best performance.

Attention	Mem	Time	NJUD [51]				NLPR [52]				DUT-RGBD [32]				STERE [54]			
	(Mb)	(s)	S_m	maxF	E_ξ	MAE	S_m	maxF	E_ξ	MAE	S_m	maxF	E_ξ	MAE	S_m	maxF	E_ξ	MAE
SA	2247	0.057	0.901	0.896	0.933	0.049	0.916	0.898	0.946	0.032	0.918	0.921	0.948	0.040	0.902	0.893	0.937	0.047
CGA	4139	0.226	0.908	0.903	0.941	0.044	0.916	0.898	0.951	0.031	0.924	0.927	0.954	0.036	0.903	0.894	0.939	0.046
CBAM[60]	2813	0.139	0.907	0.901	0.937	0.043	0.922	0.907	0.958	0.027	0.922	0.926	0.952	0.036	0.904	0.896	0.941	0.042
FGA	3033	0.066	0.906	0.902	0.936	0.045	0.927	0.912	0.961	0.025	0.926	0.927	0.954	0.034	0.904	0.896	0.94	0.042

recently published deep-learning-based models, including CTMF [21], MMCI [36], PCF [24], TANet [44], CPFP [38], DMRA [32], S^2MA [40], ICNet [31], UCNet [62], and JL-DCF [61].

The quantitative comparison in terms of the above mentioned four metrics is reported in Table 3. Since most compared models except for DMRA and S^2MA were trained on only two datasets, i.e., NJUD and NLPR, we report comparison results with all using either 2 and 3 training datasets for fair comparisons. The results show that, when using 2 training datasets, our proposed model achieves a comparable performance with the SOTA UCNet. When trained on 3 datasets, our model obviously outperforms all other methods, including all of those trained on either 2 or 3 datasets.

On the other hand, we show a qualitative model comparison of the saliency maps in Figure 6. The results show that the saliency maps of our model can not only highlight salient objects more accurately, but also recover object details more precisely (see Row III). Our model can also cope with various challenging scenarios, *e.g.*, the large statue in Row II, the very challenging relief in Row IV, and the book in row VI, where most other SOTA models fail to completely highlight the salient objects. For Rows V and VII, although the backgrounds are very cluttered, our model can successfully separate the salient objects from the backgrounds despite that other SOTA models are largely distracted by the backgrounds.

4.6. Failure Analysis

We show some common failure patterns in Figure 7. We observe that our RGB-D SOD model mainly fails in three cases. The first row of Figure 7 demonstrates that it is hard to perceive low-level (*e.g.*, color) contrast thus may incorrectly localize salient objects. The left example in the second row shows that extreme illumination condition is a challenge for our model. The right example shows that it may be distracted by cluttered backgrounds. The last row indicates that it may also fail when facing images with no obvious salient objects. All these three cases are challenging for all deep learning based SOD models. Solving these problems can be our future work.

5. Conclusion

In this paper, we have reconsidered the feature aggregation schemes for deep RGB-D SOD and proposed novel feature aggregation methods. Based on the widely used two-stream UNet architecture, we have first proposed to add early aggregation and holistic aggregation paths to propagate cross-modal information in an early stage and learn abundant feature interactions among all multi-level features. We have also proposed to cascade the top-down decoder network in UNet with a bottom-up decoder network, thus enabling to improve the high-level features with the already improved low-level features. Furthermore, we have proposed a factorized gated attention model to modulate the feature aggregation actions for each feature node with reduced computational costs and boosted model performance. Experimental results have demonstrated the effectiveness of our final RGB-D SOD model when compared with very recent state-of-the-art methods.

Table 3: Quantitative comparison of our proposed model with state-of-the-art RGB-D SOD methods. We report comparison results under two settings, i.e., training with 2 datasets (NJUD and NLPR) and training with 3 datasets (NJUD, NLPR, and DUT-RGBD). Red and blue indicate the best and the second best performance under each setting, respectively. Red means the best performance under both settings. Note that, for fair comparisons, we show the results of the JL-DCF [61] model with the VGG backbone, whose results are only reported on 6 datasets in their paper.

Dataset	Metric	Training with 2 Datasets									Training with 3 Datasets		
		CTMF [21]	MMCI [36]	PCF [24]	TANet [44]	CPFP [38]	ICNet [31]	UCNet [62]	JL-DCF [61]	Ours	DMRA [32]	S^2MA [40]	Ours*
NJUD	S_m ↑	0.849	0.858	0.877	0.878	0.878	0.894	0.897	0.897	0.908	0.886	0.894	0.906
	maxF ↑	0.845	0.852	0.872	0.874	0.877	0.891	0.895	0.899	0.901	0.886	0.889	0.902
	E_ξ ↑	0.913	0.915	0.924	0.925	0.923	0.926	0.936	0.939	0.943	0.927	0.930	0.936
[51]	MAE ↓	0.085	0.079	0.059	0.060	0.053	0.052	0.043	0.044	0.040	0.051	0.053	0.045
NLPR	S_m ↑	0.860	0.856	0.874	0.886	0.888	0.923	0.920	0.920	0.922	0.899	0.915	0.927
	maxF ↑	0.825	0.815	0.841	0.863	0.867	0.908	0.903	0.907	0.908	0.879	0.902	0.912
	E_ξ ↑	0.929	0.913	0.925	0.941	0.932	0.952	0.956	0.959	0.957	0.947	0.953	0.961
[52]	MAE ↓	0.056	0.059	0.044	0.041	0.036	0.028	0.025	0.026	0.026	0.031	0.030	0.025
RGBD135	S_m ↑	0.863	0.848	0.842	0.858	0.872	0.920	0.933	0.913	0.925	0.900	0.941	0.943
	maxF ↑	0.844	0.822	0.804	0.827	0.846	0.913	0.930	0.905	0.910	0.888	0.935	0.937
	E_ξ ↑	0.932	0.928	0.893	0.910	0.923	0.960	0.976	0.955	0.963	0.943	0.973	0.978
[17]	MAE ↓	0.055	0.065	0.049	0.046	0.038	0.027	0.018	0.026	0.018	0.030	0.021	0.016
LFSD	S_m ↑	0.796	0.787	0.794	0.801	0.828	0.868	0.864	0.833	0.860	0.847	0.837	0.879
	maxF ↑	0.791	0.771	0.779	0.796	0.826	0.871	0.864	0.840	0.867	0.856	0.835	0.881
	E_ξ ↑	0.865	0.839	0.835	0.847	0.872	0.903	0.905	0.877	0.904	0.900	0.873	0.914
[53]	MAE ↓	0.119	0.132	0.112	0.111	0.088	0.071	0.066	0.091	0.078	0.075	0.094	0.062
STERE	S_m ↑	0.848	0.873	0.875	0.871	0.879	0.903	0.903	0.894	0.897	0.886	0.890	0.904
	maxF ↑	0.831	0.863	0.860	0.861	0.874	0.898	0.899	0.889	0.887	0.886	0.882	0.896
	E_ξ ↑	0.912	0.927	0.925	0.923	0.925	0.942	0.944	0.938	0.934	0.938	0.932	0.940
[54]	MAE ↓	0.086	0.068	0.064	0.060	0.051	0.045	0.039	0.046	0.048	0.047	0.051	0.042
SSD	S_m ↑	0.776	0.813	0.841	0.839	0.807	0.848	0.865	-	0.880	0.857	0.868	0.876
	maxF ↑	0.729	0.781	0.807	0.810	0.766	0.841	0.855	-	0.871	0.844	0.848	0.852
	E_ξ ↑	0.865	0.882	0.894	0.897	0.852	0.902	0.907	-	0.926	0.906	0.909	0.915
[55]	MAE ↓	0.099	0.082	0.062	0.063	0.082	0.064	0.049	-	0.045	0.058	0.052	0.049
DUT-	S_m ↑	0.831	0.791	0.801	0.808	0.818	0.852	0.897	-	0.870	0.889	0.903	0.926
RGBD	maxF ↑	0.823	0.767	0.771	0.790	0.795	0.850	0.895	-	0.860	0.898	0.901	0.927
	E_ξ ↑	0.899	0.859	0.856	0.861	0.859	0.899	0.936	-	0.901	0.933	0.937	0.954
[32]	MAE ↓	0.097	0.113	0.100	0.093	0.076	0.072	0.043	-	0.066	0.048	0.043	0.034
SIP	S_m ↑	0.716	0.833	0.842	0.835	0.850	0.854	0.875	0.866	0.881	0.806	0.872	0.889
	maxF ↑	0.694	0.818	0.838	0.830	0.851	0.857	0.879	0.873	0.884	0.821	0.877	0.889
	E_ξ ↑	0.829	0.897	0.901	0.895	0.903	0.903	0.919	0.916	0.926	0.875	0.919	0.930
[56]	MAE ↓	0.139	0.086	0.071	0.075	0.064	0.069	0.051	0.056	0.049	0.085	0.057	0.047

Acknowledgments

This work is sponsored by Innovation Foundation for Doctor Dissertation of Northwestern Polytechnical University (CX201959) and Synergy Innovation Foundation of the University and Enterprise for Graduate Students in Northwestern Polytechnical University (XQ201910). This work is also supported in part by the National Natural Science Foundation of China under Grant 61972321.

References

[1] D. Zhang, J. Han, L. Zhao, D. Meng, Leveraging prior-knowledge for weakly supervised object detection under a collaborative self-paced curriculum learning framework 127 (4) (2019) 363–380.

[2] W. Wang, J. Shen, R. Yang, F. Porikli, Saliency-aware video object segmentation, IEEE transactions on pattern analysis and machine intelligence 40 (1) (2017) 20–33.

[3] Y. Wei, X. Liang, Y. Chen, X. Shen, M.-M. Cheng, J. Feng, Y. Zhao, S. Yan, Stc: A simple to complex framework for weakly-supervised semantic segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (11) (2017) 2314–2320.

[4] A. Chaudhry, P. K. Dokania, P. H. S. Torr, Discovering class-specific pixels for weakly-supervised semantic segmentation, in: British Machine Vision Conference, 2017.

[5] W. Wang, J. Shen, H. Ling, A deep network solution for attention and aesthetics aware photo cropping, IEEE Transactions on Pattern Analysis and Machine Intelligence 41 (7) (2019) 1531–1544.

[6] X. Fan, C. Xiang, C. Chen, P. Yang, L. Gong, X. Song, P. Nanda, X. He, Buildsensys: Reusing building sensing data for traffic prediction with cross-domain learning, IEEE Transactions on Mobile Computing (2020).

[7] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, S.-M. Hu, Global contrast based salient region detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 37 (3) (2014) 569–582.

[8] L. Wang, H. Lu, X. Ruan, M.-H. Yang, Deep networks for saliency detection via local estimation and global search, in: IEEE Conference on Computer Vision and Pattern Recognition,

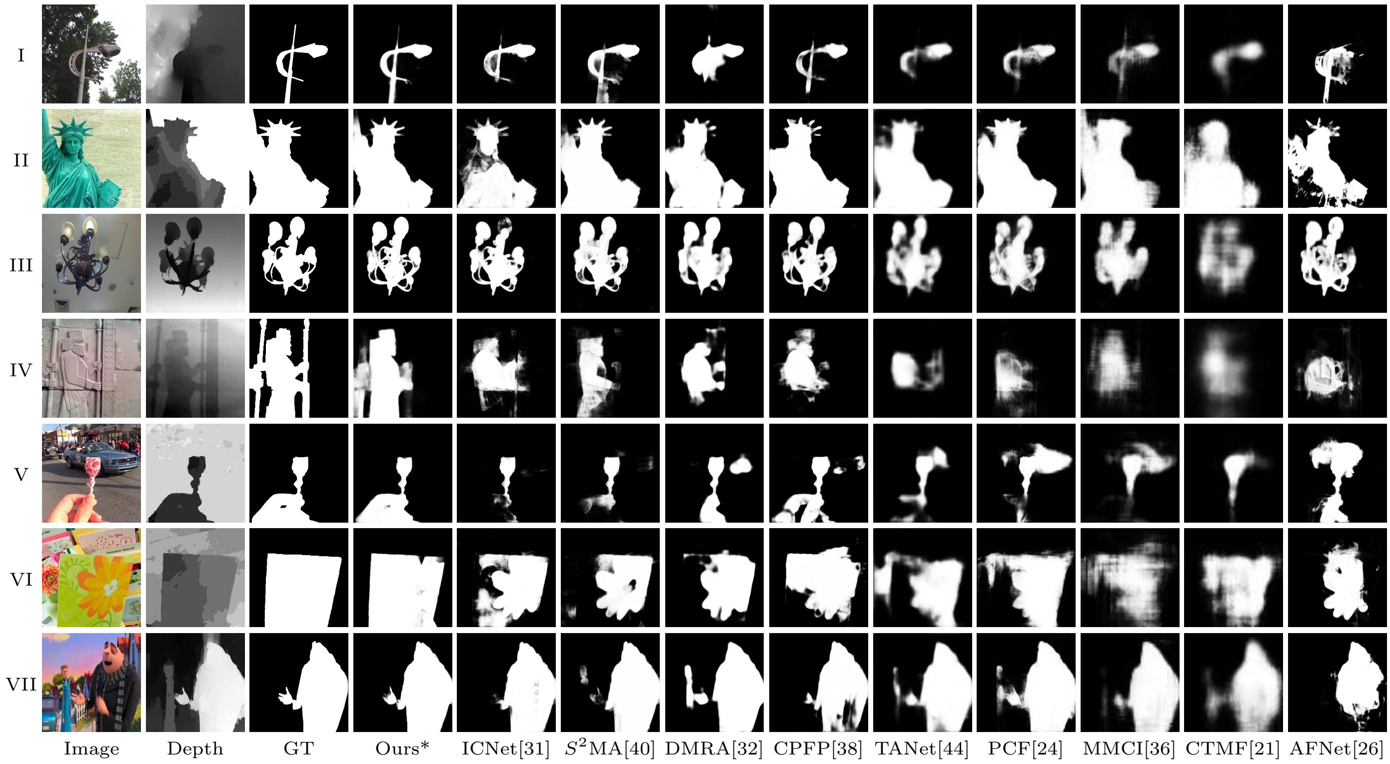


Figure 6: **Visualization of the saliency maps of our SOD model and other state-of-the-art RGB-D SOD models.**

2015, pp. 3183–3192.

[9] N. Liu, J. Han, Dhsnet: Deep hierarchical saliency network for salient object detection, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 678–686.

[10] W. Wang, J. Shen, X. Dong, A. Borji, Salient object detection driven by fixation prediction, in: IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1711–1720.

[11] J.-X. Zhao, J.-J. Liu, D.-P. Fan, Y. Cao, J. Yang, M.-M. Cheng, Egnet: Edge guidance network for salient object detection, in: IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 8779–8788.

[12] W. Wang, S. Zhao, J. Shen, S. C. Hoi, A. Borji, Salient object detection with pyramid attention and salient edges, in: IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 1448–1457.

[13] N. Liu, J. Han, M.-H. Yang, Picanet: Pixel-wise contextual attention learning for accurate saliency detection, IEEE Transactions on Image Processing (2020).

[14] N. Ouerhani, H. Hugli, Computing visual attention from scene depth, in: International Conference on Pattern Recognition, Vol. 1, IEEE, 2000, pp. 375–378.

[15] C. Lang, T. V. Nguyen, H. Katti, K. Yadati, M. Kankanhalli, S. Yan, Depth matters: Influence of depth cues on visual saliency, in: European Conference on Computer Vision, Springer, 2012, pp. 101–115.

[16] A. Ciptadi, T. Hermans, J. Rehg, An in depth view of saliency, in: British Machine Vision Conference, 2013.

[17] Y. Cheng, H. Fu, X. Wei, J. Xiao, X. Cao, Depth enhanced saliency detection method, in: International Conference on Internet Multimedia Computing and Service, ACM, 2014, p. 23.

[18] H. Song, Z. Liu, H. Du, G. Sun, O. Le Meur, T. Ren, Depth-aware salient object detection and segmentation via multiscale discriminative saliency fusion and bootstrap learning, IEEE Transactions on Image Processing 26 (9) (2017) 4204–4216.

[19] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431–3440.

[20] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, 2015, pp. 234–241.

[21] J. Han, H. Chen, N. Liu, C. Yan, X. Li, Cnns-based rgb-d saliency detection via cross-view transfer and multiview fusion, IEEE Transactions on Cybernetics 48 (11) (2017) 3171–3183.

[22] D.-P. Fan, Z. Lin, J.-X. Zhao, Y. Liu, Z. Zhang, Q. Hou, M. Zhu, M.-M. Cheng, Rethinking rgb-d salient object detection: Models, datasets, and large-scale benchmarks, arXiv preprint arXiv:1907.06781 (2019).

[23] H. Chen, Y.-F. Li, D. Su, Attention-aware cross-modal cross-level fusion network for rgb-d salient object detection, in: International Conference on Intelligent Robots and Systems, IEEE, 2018, pp. 6821–6826.

[24] H. Chen, Y. Li, Progressively complementarity-aware fusion network for rgb-d salient object detection, in: IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3051–3060.

[25] Z. Liu, S. Shi, Q. Duan, W. Zhang, P. Zhao, Salient object detection for rgb-d image by single stream recurrent convolution neural network, Neurocomputing 363 (2019) 46–57.

[26] N. Wang, X. Gong, Adaptive fusion for rgb-d salient object detection, IEEE Access 7 (2019) 55277–55284.

[27] C. Zhu, X. Cai, K. Huang, T. H. Li, G. Li, Pdnet: Prior-model guided depth-enhanced network for salient object detection, in: International Conference on Multimedia and Expo, IEEE, 2019, pp. 199–204.

[28] Y. Ding, Z. Liu, M. Huang, R. Shi, X. Wang, Depth-aware saliency detection using convolutional neural networks, Journal of Visual Communication and Image Representation 61 (2019) 1–9.

[29] G. Li, Z. Liu, L. Ye, Y. Wang, H. Ling, Cross-modal weighting network for rgb-d salient object detection, in: European Conference on Computer Vision, 2020.

[30] X. Zhou, G. Li, C. Gong, Z. Liu, J. Zhang, Attention-guided rgbd saliency detection using appearance information, Image

Image Depth GT Ours* Image Depth GT Ours*

Figure 7: **Visualization of common failure patterns.**

and Vision Computing 95 (2020) 103888.

[31] G. Li, Z. Liu, H. Ling, Icnet: Information conversion network for rgb-d based salient object detection, IEEE Transactions on Image Processing 29 (2020) 4873–4884.

[32] Y. Piao, W. Ji, J. Li, M. Zhang, H. Lu, Depth-induced multi-scale recurrent attention network for saliency detection, in: IEEE International Conference on Computer Vision, 2019, pp. 7254–7263.

[33] W. Wang, Q. Lai, H. Fu, J. Shen, H. Ling, R. Yang, Salient object detection in the deep learning era: An in-depth survey, arXiv preprint arXiv:1904.09146 (2019).

[34] L. Qu, S. He, J. Zhang, J. Tian, Y. Tang, Q. Yang, Rgbd salient object detection via deep fusion, IEEE Transactions on Image Processing 26 (5) (2017) 2274–2285.

[35] R. Shigematsu, D. Feng, S. You, N. Barnes, Learning rgb-d salient object detection using background enclosure, depth contrast, and top-down features, in: IEEE International Conference on Computer Vision Workshops, 2017, pp. 2749–2757.

[36] H. Chen, Y. Li, D. Su, Multi-modal fusion network with multi-scale multi-path and cross-modal interactions for rgb-d salient object detection, Pattern Recognition 86 (2019) 376–385.

[37] H. Song, W. Wang, S. Zhao, J. Shen, K.-M. Lam, Pyramid dilated deeper convlstm for video salient object detection, in: European Conference on Computer Vision, 2018, pp. 715–731.

[38] J.-X. Zhao, Y. Cao, D.-P. Fan, M.-M. Cheng, X.-Y. Li, L. Zhang, Contrast prior and fluid pyramid integration for rgbd salient object detection, in: IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 3927–3936.

[39] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7132–7141.

[40] N. Liu, N. Zhang, J. Han, Learning selective self-mutual attention for rgb-d saliency detection, in: IEEE Conference on Computer Vision and Pattern Recognition, 2020.

[41] Y. N. Dauphin, A. Fan, M. Auli, D. Grangier, Language modeling with gated convolutional networks, in: International Conference on Machine Learning, JMLR. org, 2017, pp. 933–941.

[42] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, T. S. Huang, Free-form image inpainting with gated convolution, in: IEEE International Conference on Computer Vision.

[43] Z. Liu, W. Zhang, P. Zhao, A cross-modal adaptive gated fusion generative adversarial network for rgb-d salient object detection, Neurocomputing 387 (2020) 210–220.

[44] H. Chen, Y. Li, Three-stream attention-aware network for rgb-d salient object detection, IEEE Transactions on Image Processing 28 (6) (2019) 2825–2835.

[45] W. Wang, J. Shen, M.-M. Cheng, L. Shao, An iterative and cooperative top-down and bottom-up inference network for salient object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 5968–5977.

[46] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: IEEE Conference on Computer Vision and Pattern Recognition, Ieee, 2009, pp. 248–255.

[47] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: International Conference on Learning Representations, 2015.

[48] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. L. Yuille, Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, IEEE Transactions on Pattern Analysis and Machine Intelligence 40 (4) (2017) 834–848.

[49] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: International Conference on Machine Learning, 2015, pp. 448–456.

[50] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, I. Polosukhin, Attention is all you need, in: Neural Information Processing Systems, 2017, pp. 5998–6008.

[51] R. Ju, L. Ge, W. Geng, T. Ren, G. Wu, Depth saliency based on anisotropic center-surround difference, in: International Conference on Image Processing, IEEE, 2014, pp. 1115–1119.

[52] H. Peng, B. Li, W. Xiong, W. Hu, R. Ji, Rgbd salient object detection: A benchmark and algorithms, in: European Conference on Computer Vision, Springer, 2014, pp. 92–109.

[53] N. Li, J. Ye, Y. Ji, H. Ling, J. Yu, Saliency detection on light field, in: IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 2806–2813.

[54] Y. Niu, Y. Geng, X. Li, F. Liu, Leveraging stereopsis for saliency analysis, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 454–461.

[55] C. Zhu, G. Li, A three-pathway psychobiological framework of salient object detection using stereoscopic technology, in: International Conference on Computer Vision Workshops, 2017, pp. 3008–3014.

[56] D.-P. Fan, Z. Lin, Z. Zhang, M. Zhu, M.-M. Cheng, Rethinking rgb-d salient object detection: Models, data sets, and large-scale benchmarks, IEEE Transactions on Neural Networks and Learning Systems (2020).

[57] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional architecture for fast feature embedding, in: ACM International conference on Multimedia, 2014, pp. 675–678.

[58] D.-P. Fan, M.-M. Cheng, Y. Liu, T. Li, A. Borji, Structure-measure: A new way to evaluate foreground maps, in: IEEE International Conference on Computer Vision, 2017, pp. 4558–4567.

[59] D.-P. Fan, C. Gong, Y. Cao, B. Ren, M.-M. Cheng, A. Borji, Enhanced-alignment measure for binary foreground map evaluation, in: International Joint Conference on Artificial Intelli-

gence, AAAI Press, 2018, pp. 698–704.

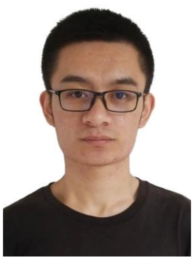
[60] S. Woo, J. Park, J.-Y. Lee, I. So Kweon, Cbam: Convolutional block attention module, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 3–19.

[61] K. Fu, D.-P. Fan, G.-P. Ji, Q. Zhao, Jl-dcf: Joint learning and densely-cooperative fusion framework for rgb-d salient object detection, in: IEEE Conference on Computer Vision and Pattern Recognition, 2020, pp. 3052–3062.

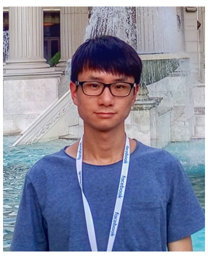
[62] J. Zhang, D.-P. Fan, Y. Dai, S. Anwar, F. S. Saleh, T. Zhang, N. Barnes, Uc-net: uncertainty inspired rgb-d saliency detection via conditional variational autoencoders, in: IEEE Conference on Computer Vision and Pattern Recognition, 2020, pp. 8582–8591.

Yuan-fang Zhang is currently pursuing his dual Ph.D. degrees with the School of Computer Science at Northwestern Polytechnical University, Shaanxi, China and Faculty of Engineering and IT, University of Technology Sydney, Australia. His current research focuses on image processing and computer vision, especially in the domains of the image enhancement, object detection and saliency detection.

Jiangbin Zheng received the Ph.D. degree from Northwestern Polytechnical University in 2002, where he is a Full Professor and Dean with School of Software. His research interests include computer graphics, computer vision and multimedia. He has published over 100 papers in the above related research area.



Long Li is a M.E. student of School of Automation at Northwestern Polytechnical University, Xi'an, China. He received the B.E. degree from Northwestern Polytechnical University in 2018. His research interests include computer vision and machine learning.



Nian Liu is a researcher with Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE. He received the Ph.D. degree, the M.S. degree, and the B.S. degree from School of Automation at Northwestern Polytechnical University, in 2020, 2015, and 2012, respectively. His research interests include computer vision and machine learning, especially on saliency detection and deep learning.

Wenjing Jia received her Ph.D. degree in Computing Science from the University of Technology Sydney (UTS) in 2007. She is currently a Senior Lecturer at the Faculty of Engineering and IT and a Core Research Member at the Global Big Data Technologies Centre, UTS. She has authored over 100 quality journal articles and conference papers. Her research interests include image/video analysis, computer vision, and pattern recognition.

Xiaochen Fan received his B.S. degree in Computer Science from the Beijing Institute of Technology, China, in 2013. He is currently a Ph.D. candidate at the School of Electrical and Data Engineering, University of Technology Sydney, Australia. His research interests include mobile edge computing, pattern recognition, deep learning, and IoT.

Chengpei Xu received his Bachelor degree in System Engineering from National University of Defence Technology, China, a Master degree in Information Technology from University of New South Wales, Australia, in 2018. His research has focused on computer vision, multimodal data fusion and artificial intelligence in e-learning.

Xiangjian He received the Ph.D. degree in Computer Science from the University of Technology Sydney (UTS), Australia in 1999. He is currently a Full Professor and the Director of the Computer Vision and Pattern Recognition Laboratory, Global Big Data Technologies Centre, UTS.