MobileSal: Extremely Efficient RGB-D Salient Object Detection

Yu-Huan Wu, Yun Liu, Jun Xu, Jia-Wang Bian, Yu-Chao Gu, and Ming-Ming Cheng

Abstract—The high computational cost of neural networks has prevented recent successes in RGB-D salient object detection (SOD) from benefiting real-world applications. Hence, this paper introduces a novel network, MobileSal, which focuses on efficient RGB-D SOD using mobile networks for deep feature extraction. However, mobile networks are less powerful in feature representation than cumbersome networks. To this end, we observe that the depth information of color images can strengthen the feature representation related to SOD if leveraged properly. Therefore, we propose an implicit depth restoration (IDR) technique to strengthen the mobile networks' feature representation capability for RGB-D SOD. IDR is only adopted in the training phase and is omitted during testing, so it is computationally free. Besides, we propose compact pyramid refinement (CPR) for efficient multi-level feature aggregation to derive salient objects with clear boundaries. With IDR and CPR incorporated, MobileSal performs favorably against state-of-the-art methods on six challenging RGB-D SOD datasets with much faster speed (450fps for the input size of 320×320) and fewer parameters (6.5M). The code is released at https://mmcheng.net/mobilesal.

Index Terms—RGB-D Salient Object Detection, Efficiency, Implicit Depth Restoration.

✦

1 INTRODUCTION

Salient object detection (SOD) aims to locate and segment the most eye-catching object(s) in natural images. It is a fundamental problem in image understanding and serves as a preliminary step for many computer vision tasks such as visual tracking [1], content-aware image editing [2], and weakly supervised learning [3]. Current SOD methods are mainly developed for RGB images [4]–[6], which are usually hindered by indistinguishable foreground and background textures. To this end, researchers resort to the easily accessible depth information as an important complement to the RGB counterpart, with promising progress in RGB-D SOD [7]–[14].

While convolutional neural networks (CNNs) have made brilliant achievements on RGB-D SOD [8]–[12], [14], their high accuracy often comes at the expense of high computational costs and large model size. This situation has prevented recent state-of-the-art methods [8], [14] from being applied to real-world applications, especially for those on mobile devices, which are depth accessible, with very limited energy overhead and computational capability. Hence, it is essential to design efficient networks for accurate RGB-D SOD. A naïve solution towards this goal is to adopt lightweight backbones such as MobileNets [16], [17] and ShuffleNets [18], [19] for deep feature extraction, instead of commonly-used cumbersome backbones like VGG [20] and ResNets [21]. The problem is that lightweight networks are usually less powerful than cumbersome networks on feature representation learning, as widely acknowledged by the research community [16]–[19]. This problem would
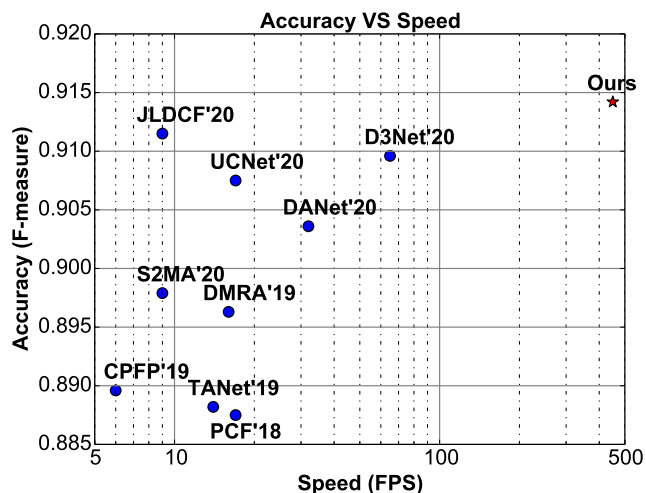
- Corresponding author: M.-M. Cheng. (E-mail: cmm@nankai.edu.cn)
- Y.-H. Wu, Y. Liu, J. Xu, Y.-C. Gu, and M.-M. Cheng are with the TKLNDST, College of Computer Science, Nankai University, Tianjin, China, 300350.
- J.-W. Bian is with the University of Adelaide.

Figure 1. Comparison with state-of-the-art methods (see references in Table 1) on the challenging NJU2K [15] dataset. Our method (MobileSal shows very competitive accuracy and much faster speed.

hinder lightweight networks from accurate RGB-D SOD performance.

To overcome this challenge, we note that the depth information of color images, if leveraged properly, can strengthen the feature representation for RGB-D SOD [8], [9]. Unlike some existing studies [11], [12] that leverage the depth information explicitly, in this paper, we propose an implicit depth restoration (IDR) technique to strengthen the feature representation learning of the lightweight backbone network so as to ensure the accuracy of RGB-D SOD in an efficient setting. More importantly, IDR is only adopted in the training phase and is omitted during testing, so it is *computationally free* during the inference stage. Specifically, we enforce our model to restore the depth map from high-

level backbone features, through which the representation learning of the lightweight backbone becomes more powerful with important supervision on the depth stream. Besides the IDR module, we propose two more components to ensure the high efficiency: i) we conduct RGB and depth information fusion only at the coarsest level, because such a small feature resolution (*i.e.*, 1/32 scale) is essential for reducing computational cost; ii) we propose a compact pyramid refinement (CPR) module to efficiently aggregate multi-scale deep features, for accurate SOD with clear boundaries.

With MobileNetV2 [17] as the backbone network, Mobile-Sal achieves 450fps on a single NVIDIA RTX 2080Ti GPU with the input size of 320×320, tens of times faster than existing RGB-D SOD methods [8], [9], [14], [22]. Extensive experiments on six challenging datasets demonstrate that MobileSal also achieves competitive performance compared with state-of-the-art methods (max F-measure of 91.4% and 91.2% on NJU2K [15] and DUTLF [9] datasets, respectively) with fewer parameters (6.5M). Such high efficiency, good accuracy, and small model size would benefit many real-world applications.

In summary, our main contributions include:

- To the best of our knowledge, MobileSal is the first to shed light upon efficient RGB-D SOD by proposing an extremely efficient network with a speed of 450fps.
- To ensure the efficiency of MobileSal on cross-modal fusion, MobileSal fuses RGB and depth information only at the coarsest level and then efficiently aggregates multi-level deep features using a compact pyramid refinement (CPR) module.
- To ensure the accuracy of MobileSal, we propose an implicit depth restoration (IDR) technique to strengthen the less powerful features learned by mobile backbone networks. This technique is also applicable to other segmentation tasks such as RGB-D semantic segmentation.

2 RELATED WORK

2.1 Salient Object Detection

Benefited from the rapid development of deep CNNs [20], [21] in recent years, CNNs-based SOD methods for RGB images [5], [6], [23]–[28] have achieved substantial progress compared with conventional methods [29]–[34]. Along this direction, much attention is paid to design various effective strategies to fuse multi-scale features generated by multi-level CNN layers [6], [23], [24], [26]. Some efforts are also spent on exploring the effectiveness of extra boundary information [24], [25], [35] or part-object relationship [36]. Detailed introductions of SOD works can refer to recent popular surveys [37]–[39]. Despite many success stories, RGB SOD is hindered by indistinguishable foreground and background textures, which can be largely alleviated by incorporating the depth information, *i.e.*, RGB-D SOD.

2.2 RGB-D Salient Object Detection

Like early SOD methods, conventional RGB-D SOD works extract hand-crafted features from RGB and depth maps and fuse them together [40]–[45]. Recently, RGB-D SOD has gained more attention and deep-learning-based RGB-D SOD has been developed rapidly [8]–[13], [46]–[50]. Typically,

Zhao *et al.* [8] proposed a contrast-prior-based network, providing strong depth enhancement. Piao *et al.* [9] proposed to refine depth via depth-induced multi-scale recurrent attention (DMRA). Huang *et al.* [49] proposed RGB-D fusion via joint cross-modal and unimodal features, providing more comprehensive RGB-D analysis. Zhang *et al.* [11] provided another perspective of uncertainty RGB-D saliency via conditional variational autoencoders. Chen *et al.* [51] first introduced 3D CNNs to RGB-D SOD, providing more abundant spatial semantics. Ji *et al.* [52] introduced a flexible depth calibration module, providing reliable complementary information for saliency models. Zhao *et al.* [47] proposed a self-supervised learning framework, which only leverages image-level annotations saving large costs from large-scale data annotations. More inspiring related works can refer to the recent survey [53].

In terms of the fusion strategy of RGB and depth information, the diagrams of RGB-D SOD architectures can be broadly divided into late fusion [42], [54], [55], early fusion [11]–[13], [51], and multi-scale fusion [8], [9], [14], [22], [46]–[49]. Late fusion appears at the end of feature extraction and only predicts the result from the fused features [42], [54], [55]. Early fusion directly concatenates the input RGB image and depth map and then derives the saliency map from such RGB-D input using the encoder-decoder network [12], hypercolumn network [11], siamese network [14], or 3D CNN [51]. Multi-scale fusion first extracts RGB and depth features separately and then aggregates RGB-D features at all levels [9], [49], at middle and high levels [56], or at middle levels [46]. Although the early fusion strategy is more efficient, multi-scale fusion is more accurate. To ensure high efficiency, our method only fuses RGB and depth features at the coarsest level in a small resolution. IDR is then applied to strengthen the feature representation learning of mobile networks in a computationally free manner.

2.3 Efficient Backbone Networks

Recent growing interests in mobile vision applications have generated a high demand for efficient CNNs. Mobile devices such as autonomous driving vehicles, robots, and smartphones only have limited computational resources, so traditional cumbersome networks, like VGG [20] and ResNets [21], are unsuitable for these platforms. To this end, some efficient networks are proposed for image classification, such as MobileNets [17], ShuffleNets [19], MnasNet [57], *etc.* There also emerge some efficient networks for semantic segmentation [58], object detection [59], and ordinary RGB SOD [60], [61]. These efficient networks are with low computational costs and thus flexible for mobile platforms. In this paper, we are the first to shed light upon efficient RGB-D SOD by adopting MobileNetV2 [17] as the backbone for deep feature extraction. Our proposed techniques aim to ensure the SOD accuracy and high efficiency simultaneously in such a lightweight setting.

3 METHODOLOGY

In this section, we first provide an overview of our method in §3.1. Then, we introduce the proposed cross-modal feature fusion scheme in §3.2, implicit depth restoration in §3.3,
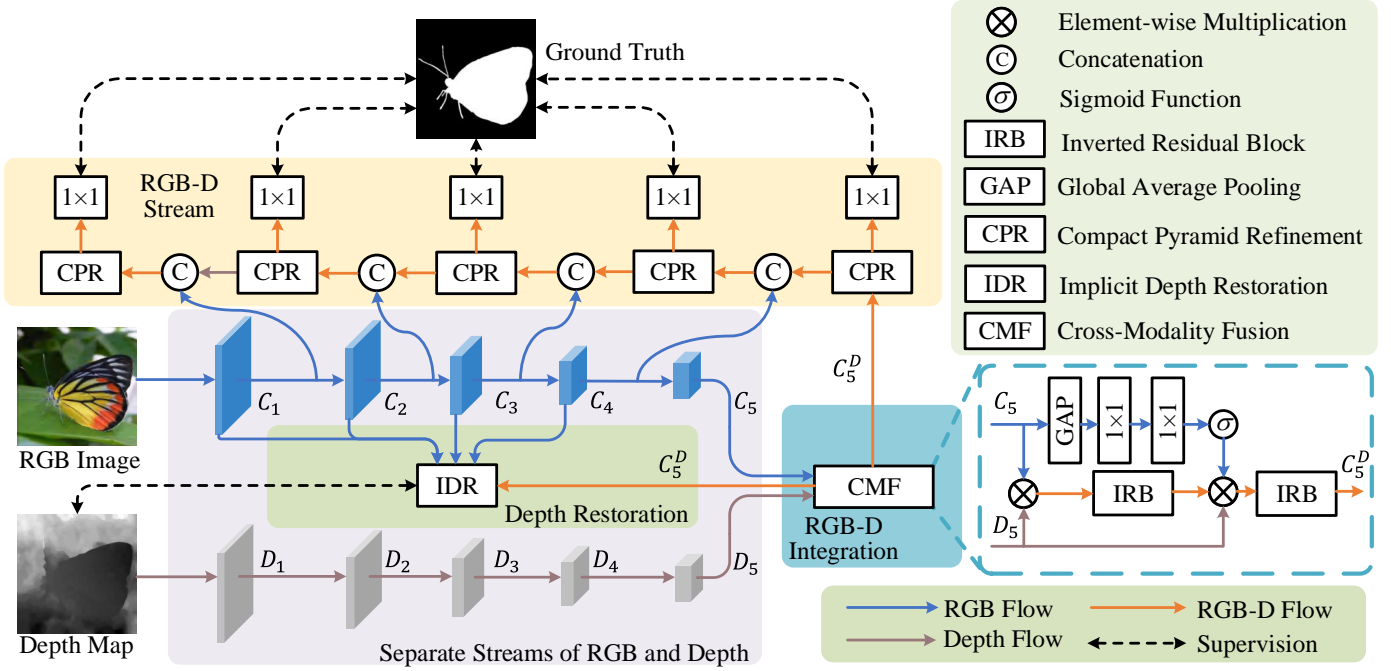
Figure 2. **The pipeline of MobileSal**. We fuse RGB and depth information only at the coarsest level and then efficiently do the multi-scale aggregation with CPRs. The IDR branch strengthens the less powerful features learned by the mobile networks in a computationally free manner.

compact pyramid refinement in §3.4. Finally, we present the hybrid loss function in §3.5.

3.1 Overview

Fig. 2 depicts the overall architecture of our method. We use RGB and depth streams for separate feature extraction.

RGB Stream. We employ MobileNetV2 [17] as the backbone of our method. To adapt it to the SOD task, we remove the global average pooling layer and the last fully-connected layer from the backbone. For the RGB stream, each stage is followed by a convolutional layer with a stride of 2, and thus feature maps are downsampled into half resolution after each stage. For convenience, we denote the output feature maps for five stages as $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4, \mathcal{C}_5$, with strides of $2, 2^2, 2^3, 2^4, 2^5$, respectively.

Depth Stream. Similar to the RGB stream, the depth stream also has five stages with the same strides. Since depth maps contain less semantic information than the corresponding RGB images, we build a lightweight depth network with fewer convolutional blocks than the RGB stream. Each stage only has two **Inverted Residual Blocks (IRB)** [17]. Such a design reduces computational complexity that accords with the goal of efficient RGB-D SOD. In each IRB, we first expand the feature map along the channel dimension by M times via a 1×1 convolution, followed by a depthwise separable 3×3 convolution [16] with the same number of input and output channels. Then, the feature channels are squeezed to $1/M$ via another 1×1 convolution. Here, each convolution is followed by Batch Normalization (BN) [62] and ReLU [63] layers, except for the last 1×1 convolution that only has a BN layer. The final output of the inverted residual block is the element-wise sum of the initial input and the output generated by the above three sequential convolutions. For the first layer in each stage, the

stride of the depthwise separable convolution is set as 2, and the number of hidden feature channels is increased if needed. The output feature maps of five stages of the depth stream are denoted as $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3, \mathcal{D}_4, \mathcal{D}_5$, the first four of which have $16, 32, 64, 96$ channels, respectively. $\mathcal{D}_5$ and $\mathcal{C}_5$ have the same number of channels and the same stride.

As shown in Fig. 2, with the outputs of the RGB and depth stream, we first fuse the extracted RGB feature $\mathcal{C}_5$ and depth feature $\mathcal{D}_5$ to generate the RGB-D feature $\mathcal{C}_5^D$. The proposed IDR technique restores the depth map from $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4, \mathcal{C}_5^D$, which is supervised by the input depth map to strengthen the feature representation learning. For saliency prediction, we design a lightweight decoder using the CPR module as the basic unit. The output of the decoder at the bottom stage is the final predicted saliency map. More details can be seen in the following sections.

3.2 Cross-Modal Fusion of RGB and Depth Features

The depth map reveals spatial cues of color images, which helps distinguish the foreground objects from the background, especially for scenarios with complicated textures. As demonstrated by previous studies [8], [9], [14], [22], [46], [48], proper RGB and depth feature fusion is essential for accurate RGB-D SOD. Our main consideration here is to ensure the high efficiency of our method. Instead of conducting fusion at multiple levels [8], [9], [14], [22], [46], [48], [64], we only fuse RGB and depth features at the coarsest level because the small feature resolution leads to low computational cost.

According to the above analyses, we only fuse the RGB feature map $\mathcal{C}_5$ and the depth feature map $\mathcal{D}_5$. We design a lightweight **Cross-Modal Fusion (CMF)** module for this purpose, as shown in Fig. 2. Intuitively, semantic information mainly exists in the RGB image. The depth map conveys the
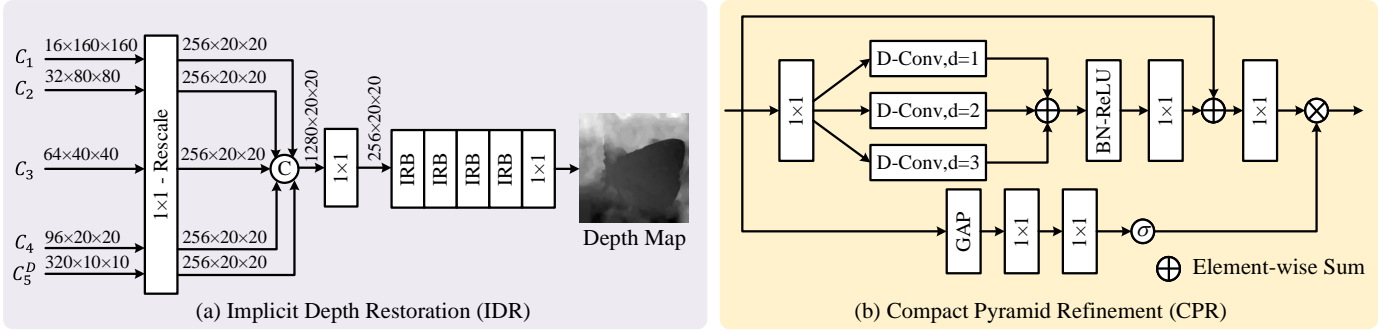
Figure 3. **Illustration of the proposed IDR and CPR.** (a) The IDR branch strengthens the less powerful features of the mobile backbone network. (b) Multi-level deep features are efficiently aggregated by the CPR module. "D-Conv" indicates depthwise separable convolution.

prior of depth-smooth regions that approximately represent the shapes and structures of complete objects or stuff. Hence, we adopt depth features like a gate to enhance RGB semantic features through multiplication, which can be viewed as a strong regularization. Note that element-wise addition or concatenation can only aggregate two feature maps by treating features equally, which is orthogonal to our goal. Experiments in §4.3 also demonstrate our hypothesis.

Specifically, we first combine the RGB and depth features with an above-mentioned IRB to derive the transited RGB-D feature maps $\mathcal{T}$, which can be formulated as

$$\mathcal{T} = \text{IRB}(\mathcal{C}_5 \otimes \mathcal{D}_5), \qquad (1)$$

where and $\otimes$ is the element-wise multiplication operator. Meanwhile, we apply a global average pooling (GAP) layer to $\mathcal{C}_5$ to get a feature vector, followed by two fully-connected layers to compute the RGB attention vector $\mathbf{v}$, like

$$\mathbf{v} = \sigma(\text{FC}_2(\text{ReLU}(\text{FC}_1(\text{GAP}(\mathcal{C}_5))))), \qquad (2)$$

in which FC and ReLU denote fully-connected and ReLU layers, respectively. The number of output channels of FC_1 and FC_2 is the same as the input. σ indicates the standard sigmoid function. With $\mathcal{T}$ and $\mathbf{v}$ computed, the multiplication of $\mathbf{v}$, $\mathcal{T}$, and $\mathcal{D}_5$ are fed into an IRB, like

$$\mathcal{C}_5^D = \text{IRB}(\mathbf{v} \otimes \mathcal{T} \otimes \mathcal{D}_5), \qquad (3)$$

where $\mathcal{C}_5^D$ indicates the output feature map of the CMF module. Note that $\mathbf{v}$ is replicated to the same shape as $\mathcal{T}$ before multiplication. Eq. (3) filters RGB semantic features again by multiplying $\mathcal{D}_5$, and the channel attention $\mathbf{v}$ is used to recalibrate the fused features. After the fusion of RGB and depth features, we can derive the backbone features, including the RGB features $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4$, and the fused RGB-D feature $\mathcal{C}_5^D$.

3.3 Implicit Depth Restoration

As widely acknowledged [16]–[19], lightweight backbone networks are less powerful in feature representation learning than cumbersome networks. To ensure the accuracy of RGB-D SOD, we consider strengthening the representation learning of mobile networks. We observe that the depth map conveys depth-smooth regions that usually represent objects, object parts, or smooth background, because intuitively, an object or a connected stuff region usually has similar depth. This observation motivates us to use the depth map

as an extra supervision source to guide the representation learning, which would help mobile networks restrain the texture changes within objects or connected stuff regions and highlight the difference among them. In this way, the contrast between salient objects and the background will be strengthened too. With this idea, we design an **Implicit Depth Restoration (IDR)** technique. Here, we use the word "implicit" because IDR is only adopted in the training phase and is omitted during testing, making it computationally free for practical deployment.

We continue by introducing how to use $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4, \mathcal{C}_5^D$ for the above auxiliary supervision. As shown in Fig. 3 (a), the pipeline of IDR is simple, *i.e.*, just concatenating multi-level feature maps and then fusing them. Specifically, we first apply a 1×1 convolution to squeeze $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4, \mathcal{C}_5^D$ to the same number of channels, *i.e.*, 256. Then, the resulting feature maps are resized to the same size as $\mathcal{C}_4$, followed by the concatenation of them. A 1×1 convolution changes the concatenated feature map from 1280 channels to 256 channels for saving computational cost. Next, four sequential IRBs are followed to fuse multi-level features so that we can obtain powerful multi-scale features. At last, a simple 1×1 convolution converts the fused feature map to a single channel. With a standard sigmoid function and bilinear upsampling, we can obtain the restored depth map with the same size as the input. The training loss of IDR adopts the well-known SSIM metric [65] to measure the structural similarity between the restored depth map $\mathcal{D}_r$ and input one $\mathcal{D}_g$, which can be written as

$$\mathcal{L}_{\text{IDR}} = 1 - \text{SSIM}(\mathcal{D}_r, \mathcal{D}_g), \qquad (4)$$

where SSIM uses the default setting. Note that the above operations are omitted during testing to make IDR free.

3.4 Compact Pyramid Refinement

It is widely accepted that high-level features in the backbone network contain semantic abstract features, while low-level features convey fine-grained details. For accurate SOD, it is essential to fully utilize both high-level and low-level features. There exists a lot of literature on this topic [8], [9], [13], [14], [22], but existing methods usually design cumbersome decoders without consideration of efficiency. Here, our decoder should not only fuse multi-level features effectively but also be efficient as much as possible.

The proposed decoder uses the **Compact Pyramid Refinement (CPR)** module as the basic unit. For efficiency, CPR

uses 1×1 and depthwise separable convolutions [16] instead of vanilla convolutions in previous methods [12]–[14], [22]. Since multi-level features exhibit multi-scale representations with the high level corresponding to the coarse scale and vice versa, multi-scale learning would be necessary for multi-level feature fusion. Hence, CPR adopts a lightweight multi-scale learning strategy to enhance such fusion. Suppose that the input of a CPR module is $\mathcal{X}$. As shown in Fig. 3 (b), CPR first applies a 1×1 convolution to expand the number of channels by M times. Then, three 3×3 depthwise separable convolutions with dilation rates of $1, 2, 3$ are connected parallel for multi-scale fusion. This can be formulated as

$$
\begin{aligned}
\mathcal{X}_1 &= \mathrm{Conv}_{1 \times 1}(\mathcal{X}), \\
\mathcal{X}_2^{d_1} &= \mathrm{Conv}_{3 \times 3}^{d_1}(\mathcal{X}_1), \\
\mathcal{X}_2^{d_2} &= \mathrm{Conv}_{3 \times 3}^{d_2}(\mathcal{X}_1), \\
\mathcal{X}_2^{d_3} &= \mathrm{Conv}_{3 \times 3}^{d_3}(\mathcal{X}_1), \\
\mathcal{X}_2 &= \mathrm{ReLU}(\mathrm{BN}(\mathcal{X}_2^{d_1} + \mathcal{X}_2^{d_2} + \mathcal{X}_2^{d_3})),
\end{aligned}
\tag{5}
$$

where d_1, d_2, and d_3 are dilation rates, *i.e.*, $1, 2, 3$ here, respectively. BN is the abbreviation of batch normalization [62]. A 1×1 convolution is used to squeeze channels to the same number as the input, *i.e.*,

$$
\mathcal{X}_3 = \mathrm{Conv}_{1 \times 1}(\mathcal{X}_2) + \mathcal{X},
\tag{6}
$$

which uses a residual connection for better optimization. The attention mechanism in Eq. (2) is applied to $\mathcal{X}$ to calculate an attention vector $\mathbf{v}'$, so that we have

$$
\mathcal{Y} = \mathbf{v}' \otimes \mathrm{Conv}_{1 \times 1}(\mathcal{X}_3).
\tag{7}
$$

Eq. (7) uses global contextual information to recalibrate the fused features.

As shown in Fig. 2, at each decoder stage, two feature maps from the top decoder and the corresponding encoder stage first reduce their numbers of channels to half using a 1×1 convolution separately. The results are then concatenated, followed by a CPR module for feature fusion. In this way, our lightweight decoder aggregates multi-level features from top to bottom.

3.5 Hybrid Loss Function

At each decoder stage, we predict the saliency map by sequentially adding a 1×1 convolution with a single channel, a sigmoid function, and bilinear upsampling to the output of the CPR module, as shown in Fig. 2. Hence, we can derive predicted saliency maps $\mathcal{P}_i(i = 1, 2, \cdots, 5)$ for five stages, respectively. Let the ground-truth saliency map be $\mathcal{G}$. The loss of each side-output can be computed as

$$
\mathcal{L}_{sal}^i = \mathrm{BCE}(\mathcal{P}_i, \mathcal{G}) + \mathrm{Dice}(\mathcal{P}_i, \mathcal{G}),
\tag{8}
$$

BCE denotes binary cross-entropy loss function:

$$
\mathrm{BCE}(\mathcal{P}_i, \mathcal{G}) = \mathcal{G} \cdot \log \mathcal{P}_i + (1 - \mathcal{G}) \cdot \log(1 - \mathcal{P}_i),
\tag{9}
$$

where "$\cdot$" indicates the dot-product operation. Dice represents the Dice loss [66]:

$$
\mathrm{Dice}(\mathcal{P}_i, \mathcal{G}) = 1 - \frac{2 \cdot \mathcal{G} \cdot \mathcal{P}_i}{||\mathcal{G}|| + ||\mathcal{P}_i||},
\tag{10}
$$

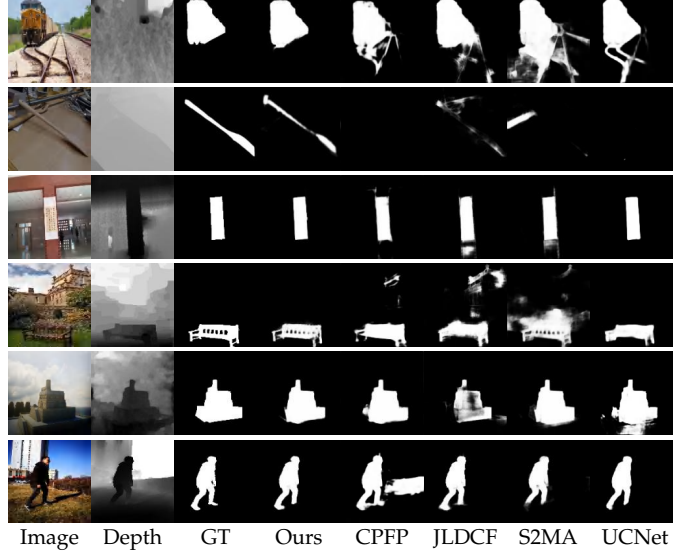Image   Depth   GT   Ours   CPFP   JLDCF   S2MA   UCNet

Figure 4. **Qualitative comparison of six challenging datasets.** The results from top to bottom are from NJU2K, DUTLF-D, NLPR, STERE, SSD, and SIP datasets, respectively.

where $|| \cdot ||$ denotes the ℓ_1 norm. With deep supervision and IDR, the training loss can be formulated as

$$
\mathcal{L} = \sum_{i=1}^{5} \mathcal{L}_{sal}^i + \lambda \cdot \mathcal{L}_{\mathrm{IDR}},
\tag{11}
$$

where λ is a balance weight. In the testing phase, $\mathcal{P}_1$ is the final predicted saliency map.

4 EXPERIMENTS

We first provide the experimental setup in §4.1. Then, we compare with state-of-the-art RGB-D SOD methods in §4.2 and conduct comprehensive ablation studies in §4.3. We also discuss the applications of IDR in §4.4.

4.1 Experimental Setup

Implementation details. We implement our network in PyTorch [70] and Jittor [71]. If not specified, we use MobileNetV2 [17] as our backbone. The M values in the depth stream, CPR, and IDR are set to 4, 4, and 6, respectively. We resize both RGB and depth images into 320×320. We use horizontal flipping and random cropping as the default data augmentation for the ablation study. After freezing the designs and parameters, we apply multi-scale training, *i.e.*, each image is resized into $[256, 288, 320]$ in training, but we keep the size of test images unchanged. We use a single RTX 2080Ti GPU for training and testing. The initial learning rate lr is 0.0001, and the batch size is 10. We train our network for 60 epochs. The *poly* learning rate policy is applied, so that the actual learning rate for each epoch cur_epoch is $(1 - \frac{cur\_epoch}{60})^{power} \times lr$, where $power$ is 0.9. The Adam optimizer [72] is used for optimizing our networks, and the momentum, weight decay, β_1, and β_2 are set as 0.9, 0.0001, 0.9, and 0.99, respectively.

Datasets. We conduct experiments on seven widely-used datasets, including NJU2K [15], DUTLF-D [9], NLPR [44], STERE [73], SSD [74], and SIP [10]. They contain 1985, 1200,

Table 1

Quantitative results on six challenging datasets. The best, second best, and third best results are highlighted in red, blue and **bold**, respectively. Our method achieves the best speed-accuracy trade-off.

Method #PubYear [Ref]		DESM 2014 [45]	LHM 2014 [44]	ACSD 2014 [15]	DCMC 2016 [67]	CTMF 2017 [54]	PCF 2018 [68]	TANet 2019 [7]	CPFP 2019 [8]	DMRA 2019 [9]	D3Net 2020 [10]	JLDCF 2020 [14]	S2MA 2020 [22]	UCNet 2020 [11]	DANet 2020 [12]	BiANet 2021 [69]	MobileSal (Ours)
Params (M)		-	-	-	-	-	133.4	232.4	69.5	59.7	43.2	137.0	86.7	33.3	26.7	49.6	6.5
Speed (FPS)		-	-	1	-	8	17	14	6	16	65	9	9	17	32	50	450
NJU2K	F_β^{max} ↑	0.767	0.703	0.749	0.759	0.857	0.887	0.888	0.890	0.896	**0.910**	0.912	0.898	0.908	0.904	0.908	0.914
	MAE ↓	0.286	0.204	0.200	0.171	0.085	0.059	0.060	0.053	0.051	0.047	0.041	0.054	**0.043**	0.047	0.044	0.041
	S_α ↑	0.671	0.515	0.708	0.686	0.849	0.877	0.878	0.878	0.886	0.900	**0.902**	0.894	0.897	0.897	0.904	0.905
	E_ξ^{max} ↑	0.807	0.738	0.814	0.805	0.913	0.924	0.925	0.923	0.927	0.939	**0.944**	0.930	0.936	0.936	0.941	0.942
	Rank ↓	15	15	13	13	12	11	10	9	8	4	2	7	5	6	3	1
DUTLF-D	F_β^{max} ↑	0.728	0.652	0.212	0.419	0.811	0.782	0.804	0.740	0.887	0.748	0.884	0.882	0.836	0.869	**0.885**	0.912
	MAE ↓	0.293	0.162	0.320	0.232	0.095	0.100	0.092	0.100	0.053	0.099	0.053	0.054	0.064	0.054	**0.048**	0.041
	S_α ↑	0.659	0.568	0.361	0.499	0.831	0.801	0.808	0.749	0.888	0.775	**0.906**	0.903	0.863	0.889	0.906	0.896
	E_ξ^{max} ↑	0.800	0.734	0.590	0.654	0.899	0.856	0.861	0.811	0.933	0.834	**0.943**	0.937	0.904	0.931	0.946	0.950
	Rank ↓	13	14	16	15	8	10	9	12	4	11	3	5	7	6	2	1
NLPR	F_β^{max} ↑	0.680	0.693	0.664	0.706	0.841	0.863	0.877	0.888	0.888	0.907	**0.925**	0.910	0.915	0.907	0.921	0.916
	MAE ↓	0.316	0.104	0.163	0.112	0.056	0.044	0.041	0.036	0.031	0.030	0.022	0.030	**0.025**	0.031	0.024	0.025
	S_α ↑	0.573	0.631	0.684	0.729	0.860	0.874	0.886	0.888	0.899	0.912	**0.925**	0.915	**0.920**	0.909	0.927	0.920
	E_ξ^{max} ↑	0.808	0.763	0.800	0.795	0.929	0.925	0.941	0.932	0.947	0.953	**0.963**	0.953	0.956	0.949	0.962	0.961
	Rank ↓	16	14	15	13	12	11	10	9	8	6	1	5	4	7	2	3
STERE	F_β^{max} ↑	0.738	0.752	0.682	0.789	0.848	0.875	0.878	0.889	0.895	0.904	**0.913**	0.895	**0.908**	0.895	**0.908**	0.906
	MAE ↓	0.301	0.172	0.197	0.148	0.086	0.064	0.060	0.051	0.047	0.046	**0.040**	0.051	**0.039**	0.048	0.042	**0.041**
	S_α ↑	0.642	0.562	0.692	0.731	0.848	0.875	0.871	0.879	0.886	0.899	**0.903**	0.890	**0.903**	0.892	**0.904**	0.903
	E_ξ^{max} ↑	0.811	0.771	0.806	0.819	0.912	0.925	0.923	0.925	0.938	0.938	**0.947**	0.932	**0.944**	0.930	**0.944**	0.940
	Rank ↓	14	14	14	13	12	10	10	9	6	5	2	7	1	7	3	4
SSD	F_β^{max} ↑	0.720	0.633	0.709	0.755	0.744	0.833	0.835	0.801	0.858	0.856	0.835	**0.878**	**0.881**	**0.878**	0.870	0.863
	MAE ↓	0.313	0.195	0.204	0.169	0.098	0.062	0.063	0.082	0.059	0.059	0.085	0.053	**0.049**	**0.050**	**0.052**	0.052
	S_α ↑	0.602	0.566	0.675	0.704	0.776	0.841	0.839	0.807	0.857	0.857	0.822	**0.868**	0.866	**0.869**	**0.870**	0.862
	E_ξ^{max} ↑	0.769	0.717	0.785	0.786	0.865	0.894	0.897	0.852	**0.906**	0.898	0.898	**0.909**	0.907	0.907	0.907	0.914
	Rank ↓	15	16	14	13	12	9	8	11	7	6	10	3	1	3	2	5
SIP	F_β^{max} ↑	0.720	0.634	0.788	0.680	0.717	0.860	0.849	0.869	0.852	0.880	**0.903**	0.891	0.896	**0.900**	0.895	**0.898**
	MAE ↓	0.303	0.184	0.175	0.186	0.140	0.071	0.075	0.064	0.086	0.063	**0.049**	0.057	**0.051**	0.054	**0.051**	0.053
	S_α ↑	0.616	0.511	0.732	0.683	0.716	0.842	0.835	0.850	0.806	0.860	**0.880**	0.872	0.875	**0.878**	**0.884**	0.873
	E_ξ^{max} ↑	0.770	0.716	0.838	0.743	0.829	0.901	0.895	0.903	0.875	0.909	**0.925**	0.919	0.919	**0.921**	**0.928**	0.916
	Rank ↓	14	16	12	15	13	9	10	8	11	7	1	6	4	3	2	5

Table 2

CPU inference time of different methods. Default input size of each method is applied to test CPU inference time.

Method	Ours	JLDCF [14]	UCNet [11]
Input Size	320×320	320×320	352×352
Inference Time (ms)	**43 (1×)**	7246 (150×)	784 (18×)
Method	D3Net [10]	S2MA [22]	DMRA [9]
Input Size	224×224	256×256	256×256
Inference Time (ms)	677 (15×)	3049 (70×)	2381 (55×)

Table 3

Quantitative comparisons of restored depth maps by IDR with different scales of input depth maps. The ground-truth depth (GT) recovered from different resolutions using different interpolation methods are compared.

Setting	(a)	(b)	(c)	(d)
Scale	1/16	1/8	1/32	1/8
Data	IDR	GT	GT	GT
Up Type	Bilinear	Bilinear	Bilinear	Nearest
PSNR	22.86	30.17	22.55	24.27
SSIM	.8687	.9194	.8445	.8170

Table 4

Ablation study for the RGB-D fusion and IDR branch. Note that the variants of No. 6 and 12 (in **bold**) fuse RGB and depth features only at the coarsest feature level.

No.	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_3$	$\mathcal{C}_4$	$\mathcal{C}_5$	IDR	F_β^{max}	MAE
1	✔	✔	✔	✔	✔	✔	0.899	0.052
2	✔	✔				✔	0.894	0.050
3	✔	✔	✔			✔	0.897	0.047
4		✔	✔	✔		✔	0.902	0.048
5			✔	✔	✔	✔	0.902	0.046
6					✔	✔	**0.906**	**0.045**
7	✔	✔	✔	✔	✔		0.895	0.047
8	✔	✔					0.892	0.049
9	✔	✔	✔				0.896	0.048
10		✔	✔	✔			0.895	0.048
11			✔	✔	✔		0.898	0.048
12					✔		**0.896**	**0.047**
13							0.887	0.052

1000, 1000, 80, 927 images, respectively. Following [8], [11], [12], [22], we use 1500 images of NJU2K [15] and 700 images of NLPR [44] for training, and the other 485 images of NJU2K [15] and 300 images of NLPR [44] for testing. Except for DUTLF-D [9], other datasets are directly used for testing. On the DUTLF-D [9] dataset, we follow [9], [12] to use 800 images for training and the other 400 images for testing.

Evaluation metrics. Following recent works [7], [8], [11], [68], we adopt two widely-used metrics for evaluation. The first is F-measure F_β, where β is set as 0.3 to emphasize the significance of the precision, as suggested by [8], [9], [12], [75]. We compute the maximum F_β as F_β^{max} under different thresholds. Higher F_β^{max} indicates better performance. The second is the Mean Absolute Error (MAE), which is the lower, the better. We also report scores of recently proposed S-measure S_α [76] and maximum E-measure E_ξ^{max} [77] under different thresholds for reference. We follow the official paper to compute S_α and E_ξ^{max}. We calculate the overall rank of each method on each dataset based on the above four metrics. Besides, we report the number of parameters and running time of each method for efficiency analysis.

Table 5
Comparison of RGB-D fusion strategies. The results of default fusion strategy are with **bold** fonts.

Metric	Single Stream		Two Streams	
	IDR ✔	IDR ✘	IDR ✔	IDR ✘
$F_\beta^{\max}$	0.900	0.894	**0.906**	**0.896**
MAE	0.048	0.051	**0.045**	**0.047**

Figure 5. **Visual comparisons of restored depth maps by IDR with different scales of input depth maps.** Results of the last 3 columns have been upsampled with bilinear interpolation to match the size of the input depth map.

4.2 Comparison to state-of-the-art methods

We first compare our method with 15 recent state-of-the-art methods published on six widely-used datasets. Most methods are based on VGG-16 [20], except the ResNet-101-based JLDCF and ResNet-50-based UCNet. The saliency maps of other methods are from their released results if provided, otherwise they are computed by their released models.

Quantitative comparison. Table 1 shows the results. Our method runs at 450fps and only has 6.5M parameters. Other methods are much slower and heavier than our method. For example, JLDCF [14] is 50× slower and has 20× more parameters. UCNet is 26× slower and has 5.5× more parameters. Besides, our method outperforms other methods on NJU2K [15] and DUTLF-D [9] datasets, and ranks from 3rd to 5th on the other 4 datasets. The above results demonstrate the high efficiency and accuracy of our method.

Qualitative comparison. Fig. 4 shows the results. Due to the limited space, here we only compare our method with CPFP [8], JLDCF [14], S2MA [22], and UCNet [11] on all involved datasets. Our method can work well on several kinds of complex scenarios with noisy depth information, while others may fail in such scenarios.

CPU inference time. We test the inference time for different methods in a single core of Intel i7 8700K@3.7GHz CPU. The results are shown in Table 2. While the inference time of other state-of-the-art methods (677 ∼ 7246 ms) is far

from the bar of real-time speed (∼50 ms), the CPU inference time of our method can achieve a real-time speed of 43 ms for each RGB-D input.

4.3 Ablation Study

We evaluate each proposed component on the test split of the NJU2K [15] dataset. $F_\beta^{\max}$ and MAE are applied as the primary metrics. The results are analyzed below.

Different RGB-D fusions. Table 4 shows the results for the RGB-D fusion at different stages. When trained with the IDR branch, fusing RGB and depth features at the coarsest level results in the best performance (No. 6). Fusing RGB and depth features in the last three levels results in the best performance when trained without the IDR branch (No. 11). The IDR branch substantially improves the performance in most cases (No. 3 - 6), and only MAE (fusing low-level features) degrades (No. 1, 2). This validates the efficacy of the combination of our proposed fusion strategy and the IDR branch. We also compare the used fusion strategy with the early fusion strategy, which concatenates the input RGB image and depth map at the input stage. Although the latter strategy is more efficient, our initial fusion strategy significantly outperforms it (Table 5). Hence, To ensure accuracy and efficiency, we fuse RGB and depth features at the coarsest level.

Saved time for RGB-D fusion. In MobileSal, we conduct RGB and depth information fusion only at the coarsest level. Conducting RGB-D feature fusion of all levels will result in 260fps for our method. Hence, we will save 42% ($1 - \frac{260}{450}$) time. In other words, fusing only the coarsest will speed up our method by 73% ($\frac{450}{260} - 1$).

Depth restoration quality. We explore the restored depth quality of IDR with the widely-used PSNR and SSIM [65] metrics. The scale of the restored depth map in IDR is 1/16 of the input depth map. For comparison, we evaluate the quality of the nearest and bilinear interpolation of the input depth map of 1/8 scale. As the IDR branch receives depth features of the 1/32 scale, we also report the quality of the bilinear interpolation of depth GT with the 1/32 scale. Results are shown in Table 3. We observe that the restored depth map is closer to the input depth map than the upsampled 1/32 GT. We also conduct visual comparisons of the above settings in Fig. 5. One can see that the restored depth maps from IDR keep good restoration quality and are with less noise than the upsampled 1/8 GT.

Loss selection in IDR. As introduced in §3.3, instead of using the trivial L1/L2 loss, the SSIM metric [65] is chosen as the loss of IDR. We validate this design via training our MobileSal with L1/L2/SSIM loss. We find that IDR with the L1/L2 loss can enhance MobileSal by 0.5%/0.7% in terms of the $F_\beta^{\max}$, and is 0.5%/0.3% lower than IDR with the SSIM loss. This is because SSIM loss provides the structural similarity rather than the simple point-to-point error computed by the L1/L2 loss. Based on the above discussion, we choose the SSIM loss as the supervision of IDR.

The λ coefficient in the loss function. λ decides the loss weight of the depth restoration loss, as described in Eq. (11). We conduct experiments on our method with different λ settings. The results are shown in Table 6. The IDR branch

Table 6
Ablation study for λ coefficient selection.

No.	1	2	3	4	5	6
λ	0	0.1	0.3	0.5	1.0	2.0
$F_\beta^{\max}$	0.896	0.902	**0.906**	0.902	0.903	0.902
MAE	0.047	0.046	**0.045**	0.046	0.044	0.046

Table 7
Efficacy of CMF. "RGB" denotes the RGB backbone with the RGB input. "Depth" indicates the network with the depth stream and depth input.

No.	RGB	CPR	Depth	CMF	$F_\beta^{\max}$	MAE
1	✔				0.852	0.068
2	✔	✔			0.887	0.052
3	✔	✔	✔		0.894	0.048
4	✔	✔	✔	✔	**0.906**	**0.045**

Table 8
Comparison of different operations for the initial RGB-D fusion in Eq. (1). "Multiplication" and "Addition" operations are element-wise. Concatenation operation is along channels. The results of the default fusion strategy are with bold fonts.

Operation	Multiplication		Addition		Concatenation	
Metric	$F_\beta^{\max}$	MAE	$F_\beta^{\max}$	MAE	$F_\beta^{\max}$	MAE
Results	**0.906**	**0.045**	0.897	0.048	0.900	0.046

brings substantial improvement to the robustness of our method with different λ. Since the third column achieves the best performance, we adopt $\lambda = 0.3$ as the default setting for training our network.

Depth information and the CMF module. The results in Table 7 demonstrate the effects of depth information and the CMF module. Results with depth input are trained with the IDR branch. The providence of depth maps without the CMF module only uses an element-wise multiplication for RGB-D fusion. The results show that the depth information is very helpful for RGB-D SOD even with a very simple operation. More specifically, we also observe substantial improvement with the CMF module.

Operation for initial RGB-D fusion in the CMF module. As formulated in Eq. (1), an element-wise multiplication is applied to the initial RGB-D fusion. To validate the effectiveness of the selected operation, we compare this design with the widely-used element-wise addition. The results are presented in Table 8. Element-wise multiplication largely outperforms element-wise addition and concatenation by 0.9% and 0.6% in terms of the maximum F-measure, suggesting the superiority of element-wise multiplication in the initial RGB-D fusion of the CMF module.

Final fusion strategy of the CMF module. As described in Eq. (3), both the RGB attention vector $\mathbf{v}$ and RGB-D feature $\mathcal{T}$ join the fusion with depth feature D_5. We validate the effectiveness of the above design via removing $\mathbf{v}$ or $\mathcal{T}$. Experimental results are shown in Table 9. We observe a large performance degradation after removing $\mathcal{T}$. This is because in the feature fusion of Eq. (3), only $\mathcal{T}$ contains spatial-wise RGB features. Besides, $\mathbf{v}$ is an attention vector obtained by RGB features, and D_5 is the pure depth feature. Removing $\mathbf{v}$ also affects the performance a bit because it provides the channel-wise RGB attention in the RGB-D fusion.

Table 9
Analysis of the fusion of CMF in Eq. (3). No. 1: trivial RGB-D fusion via element-wise multiplication; No. 2-3: variants of removing $\mathbf{v}$ or $\mathcal{T}$ in Eq. (3); No. 4: the default design.

No.	Depth	CMF	$\mathbf{v}$	$\mathcal{T}$	$F_\beta^{\max}$	MAE
1	✔				0.894	0.048
2	✔	✔	✔		0.895	0.049
3	✔	✔		✔	0.902	0.046
4	✔	✔	✔	✔	**0.906**	**0.045**

Table 10
Ablation study for the dilation rates of CPR. "D Rates" means the dilation rates of each depthwise separable convolution.

No.	1	2	3	4	5	6
D Rates	1, 2, 3	1	2	3	1, 3, 6	1, 4, 8
$F_\beta^{\max}$	**0.906**	0.900	0.892	0.897	0.903	0.901
MAE	**0.045**	0.047	0.048	0.047	0.046	0.048

Compact pyramid refinement. Table 10 shows the results for CPR, where different dilation strategies are used. We test the default setting (No. 1), single convolution with different dilation rates (No. 2 - 4), and convolutions with sparse combinations of dilation rates (No. 5, 6). The default setting with compact dilation rates (1, 2, 3) significantly outperforms other settings, demonstrating the efficacy of CPR.

Hybrid loss. To validate the effectiveness of the Dice loss, we test the performance only trained with the binary cross-entropy loss. We find that adding Dice loss supervision will improve the MAE performance by 0.1% $\sim$ 0.2% via providing high contrast, but will not affect the performance of $F_\beta^{\max}$.

4.4 Application of IDR to other tasks

Our proposed IDR freely strengthens the feature representations of the backbone network, given the RGB-D input in the inference stage. It is unbound to RGB-D SOD that is a segmentation task aiming at deriving a saliency map. To show the potential of IDR on other tasks, we evaluate the performance gain of IDR in the RGB-D semantic segmentation. The goal of this task, *i.e.*, assigning each pixel with semantic labels, is similar to RGB-D SOD that predicts saliency probability of each pixel.

Experimental setup. We select two recent state-of-the-art representatives [78], [79] as the baselines. We use the official code provided by the authors to implement our ideas. Following [78], [79], we conduct our experiments on the NYUDv2 dataset [80], which consists of 1449 RGB images with corresponding depth maps and pixel-level semantic labels containing 40 semantic classes. This dataset has 795 and 654 images for training and testing, respectively. The training and testing settings follow the official papers [78], [79]. Similar to MobileSal, for [79], RGB features re-calibrated by the depth features for each stage are fed into the IDR branch. However, for [78], different from MobileSal, the output features of the first four stages are the input of the IDR branch because the features of the last stage in [78] are directly used for the prediction of semantic segmentation.

Evaluation metrics. Following [78], [79], mean IoU (mIoU), is used as the primary evaluation metric. We also

Table 11
Effect of IDR for the methods of RGB-D semantic segmentation.
Higher values of all metrics indicate better performance.

Method	SGNet [78]			Chen *et al.* [79]		
Metrics (%)	mIoU	Acc	mAcc	mIoU	Acc	mAcc
Baseline	49.6	75.6	61.9	51.4	77.1	62.9
+ IDR	50.5(+0.9)	76.3	62.2	52.2(+0.8)	77.3	64.0

report the results of pixel accuracy (Acc) and mean accuracy (mAcc) for reference. Please refer to [78] for more details about the computation of the above three metrics.

Experimental results. We show the results in Table 11. With the computationally-free IDR incorporated, we observe that the performance of both methods [78], [79] has a large improvement in terms of the mIoU metric. This suggests that the idea of IDR is also applicable and powerful for RGB-D semantic segmentation without any extra inference cost.

5 CONCLUSION

We propose a new method, MobileSal, which aims at efficient RGB-D SOD. Unlike other accurate RGB-D SOD methods, we are the first to shed light upon efficient RGB-D SOD by proposing an extremely efficient network MobileSal with a speed of 450fps. With less powerful features provided by the mobile backbone network, we propose the implicit depth restoration (IDR) technique to strengthen the less powerful features learned by mobile backbone networks. We perform ablation study for the proposed techniques in MobileSal and demonstrate their effectiveness. We conduct experimental comparisons with state-of-the-art methods on six popular benchmarks. The results show that MobileSal performs favorably against state-of-the-art methods, with fewer parameters and much faster speed. In terms of CPU inference time, our method is 15∼150× faster. Our method can serve as a strong baseline for future efficient RGB-D SOD research.

6 ACKNOWLEDGMENT

REFERENCES

[1] S. Hong, T. You, S. Kwak, and B. Han, "Online tracking by learning discriminative saliency map with convolutional neural network," in *Int. Conf. Mach. Learn.*, 2015, pp. 597–606.

[2] W. Wang, J. Shen, and H. Ling, "A deep network solution for attention and aesthetics aware photo cropping," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1531–1544, 2018.

[3] Y. Liu, Y.-H. Wu, P.-S. Wen, Y.-J. Shi, Y. Qiu, and M.-M. Cheng, "Leveraging instance-, image-and dataset-level information for weakly supervised instance segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020.

[4] P. Zhang, D. Wang, H. Lu, H. Wang, and X. Ruan, "Amulet: Aggregating multi-level convolutional features for salient object detection," in *Int. Conf. Comput. Vis.*, 2017, pp. 202–211.

[5] Y. Pang, X. Zhao, L. Zhang, and H. Lu, "Multi-scale interactive network for salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 9413–9422.

[6] X. Zhao, Y. Pang, L. Zhang, H. Lu, and L. Zhang, "Suppress and balance: A simple gated network for salient object detection," in *Eur. Conf. Comput. Vis.*, 2020, pp. 35–51.

[7] H. Chen and Y. Li, "Three-stream attention-aware network for RGB-D salient object detection," *IEEE Trans. Image Process.*, vol. 28, no. 6, pp. 2825–2835, 2019.

[8] J. Zhao, Y. Cao, D.-P. Fan, X.-Y. Li, L. Zhang, and M.-M. Cheng, "Contrast prior and fluid pyramid integration for RGBD salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 3927–3936.

[9] Y. Piao, W. Ji, J. Li, M. Zhang, and H. Lu, "Depth-induced multi-scale recurrent attention network for saliency detection," in *Int. Conf. Comput. Vis.*, 2019, pp. 7254–7263.

[10] D.-P. Fan, Z. Lin, Z. Zhang, M. Zhu, and M.-M. Cheng, "Rethinking rgb-d salient object detection: Models, data sets, and large-scale benchmarks," *IEEE Trans. Neur. Net. Learn. Syst.*, vol. 32, no. 5, pp. 2075–2089, 2020.

[11] J. Zhang, D.-P. Fan, Y. Dai, S. Anwar, F. Saleh, S. Aliakbarian, and N. Barnes, "Uncertainty inspired RGB-D saliency detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2021.

[12] X. Zhao, L. Zhang, Y. Pang, H. Lu, and L. Zhang, "A single stream network for robust and real-time RGB-D salient object detection," in *Eur. Conf. Comput. Vis.*, 2020, pp. 646–662.

[13] C. Li, R. Cong, Y. Piao, Q. Xu, and C. C. Loy, "RGB-D salient object detection with cross-modality modulation and selection," in *Eur. Conf. Comput. Vis.*, 2020, pp. 225–241.

[14] K. Fu, D.-P. Fan, G.-P. Ji, Q. Zhao, J. Shen, and C. Zhu, "Siamese network for RGB-D salient object detection and beyond," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2021.

[15] R. Ju, L. Ge, W. Geng, T. Ren, and G. Wu, "Depth saliency based on anisotropic center-surround difference," in *Int. Conf. Image Process.*, 2014, pp. 1115–1119.

[16] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[17] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 4510–4520.

[18] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 6848–6856.

[19] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet v2: Practical guidelines for efficient cnn architecture design," in *Eur. Conf. Comput. Vis.*, 2018, pp. 116–131.

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Int. Conf. Learn. Represent.*, 2015.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 770–778.

[22] N. Liu, N. Zhang, and J. Han, "Learning selective self-mutual attention for RGB-D saliency detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 13 756–13 765.

[23] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. H. Torr, "Deeply supervised salient object detection with short connections." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 4, p. 815, 2018.

[24] X. Qin, Z. Zhang, C. Huang, C. Gao, M. Dehghan, and M. Jagersand, "BASNet: Boundary-aware salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 7479–7489.

[25] J.-J. Liu, Q. Hou, M.-M. Cheng, J. Feng, and J. Jiang, "A simple pooling-based design for real-time salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 3917–3926.

[26] P. Zhang, W. Liu, H. Lu, and C. Shen, "Salient object detection with lossless feature reflection and weighted structural loss," *IEEE Trans. Image Process.*, vol. 28, no. 6, pp. 3048–3060, 2019.

[27] Y.-H. Wu, Y. Liu, L. Zhang, W. Gao, and M.-M. Cheng, "Regularized densely-connected pyramid network for salient instance segmentation," *IEEE Trans. Image Process.*, vol. 30, pp. 3897–3907, 2021.

[28] Y.-H. Wu, Y. Liu, L. Zhang, M.-M. Cheng, and B. Ren, "EDN: Salient object detection via extremely-downsampled network," *arXiv preprint arXiv:2012.13093*, 2021.

[29] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, "Saliency detection via graph-based manifold ranking," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 3166–3173.

[30] M.-M. Cheng, J. Warrell, W.-Y. Lin, S. Zheng, V. Vineet, and N. Crook, "Efficient salient region detection with soft image abstraction," in *Int. Conf. Comput. Vis.*, 2013, pp. 1529–1536.

[31] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S.-M. Hu, "Global contrast based salient region detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 569–582, 2014.

[32] J. Wang, H. Jiang, Z. Yuan, M.-M. Cheng, X. Hu, and N. Zheng, "Salient object detection: A discriminative regional feature integration approach," *Int. J. Comput. Vis.*, vol. 123, no. 2, pp. 251–268, 2017.

[33] X. Li, H. Lu, L. Zhang, X. Ruan, and M.-H. Yang, "Saliency detection via dense and sparse reconstruction," in *Int. Conf. Comput. Vis.*, 2013, pp. 2976–2983.

[34] L. Zhang, J. Ai, B. Jiang, H. Lu, and X. Li, "Saliency detection via absorbing markov chain with learnt transition probability," *IEEE Trans. Image Process.*, vol. 27, no. 2, pp. 987–998, 2017.

[35] R. Wu, M. Feng, W. Guan, D. Wang, H. Lu, and E. Ding, "A mutual learning method for salient object detection with intertwined multi-supervision," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 8150–8159.

[36] Y. Liu, D. Zhang, Q. Zhang, and J. Han, "Part-object relational visual saliency," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2021.

[37] J. Han, D. Zhang, G. Cheng, N. Liu, and D. Xu, "Advanced deep-learning techniques for salient and category-specific object detection: a survey," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 84–100, 2018.

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

[39] W. Wang, Q. Lai, H. Fu, J. Shen, H. Ling, and R. Yang, "Salient object detection in the deep learning era: An in-depth survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2021.

[40] C. Lang, T. V. Nguyen, H. Katti, K. Yadati, M. Kankanhalli, and S. Yan, "Depth matters: Influence of depth cues on visual saliency," in *Eur. Conf. Comput. Vis.*, 2012, pp. 101–115.

[41] J. R. Arridhana Ciptadi, Tucker Hermans, "An in depth view of saliency," in *Proceedings of the British Machine Vision Conference*. BMVA Press, 2013.

[42] K. Desingh, K. M. Krishna, D. Rajan, and C. Jawahar, "Depth really matters: Improving visual salient region detection with depth," in *Proceedings of the British Machine Vision Conference*. BMVA Press, 2013.

[43] X. Fan, Z. Liu, and G. Sun, "Salient region detection for stereoscopic images," in *International Conference on Digital Signal Processing*. IEEE, 2014, pp. 454–458.

[44] H. Peng, B. Li, W. Xiong, W. Hu, and R. Ji, "RGBD salient object detection: a benchmark and algorithms," in *Eur. Conf. Comput. Vis.*, 2014, pp. 92–109.

[45] Y. Cheng, H. Fu, X. Wei, J. Xiao, and X. Cao, "Depth enhanced saliency detection method," in *International Conference on Internet Multimedia Computing and Service*, 2014, pp. 23–27.

[46] Y. Pang, L. Zhang, X. Zhao, and H. Lu, "Hierarchical dynamic filtering network for RGB-D salient object detection," in *Eur. Conf. Comput. Vis.*, 2020, pp. 235–252.

[47] X. Zhao, Y. Pang, L. Zhang, H. Lu, and X. Ruan, "Self-supervised representation learning for RGB-D salient object detection," *arXiv preprint arXiv:2101.12482*, 2021.

[48] S. Chen and Y. Fu, "Progressively guided alternate refinement network for RGB-D salient object detection," in *Eur. Conf. Comput. Vis.*, 2020, pp. 520–538.

[49] N. Huang, Y. Liu, Q. Zhang, and J. Han, "Joint cross-modal and unimodal features for RGB-D salient object detection," *IEEE Trans. Multimedia*, vol. 23, pp. 2428–2441, 2021.

[50] L. Qu, S. He, J. Zhang, J. Tian, Y. Tang, and Q. Yang, "RGBD salient object detection via deep fusion," *IEEE Trans. Image Process.*, vol. 26, no. 5, pp. 2274–2285, 2017.

[51] Q. Chen, Z. Liu, Y. Zhang, K. Fu, Q. Zhao, and H. Du, "RGB-D salient object detection via 3d convolutional neural networks," in *AAAI Conf. Artif. Intell.*, 2021.

[52] W. Ji, J. Li, S. Yu, M. Zhang, Y. Piao, S. Yao, Q. Bi, K. Ma, Y. Zheng, H. Lu *et al.*, "Calibrated rgb-d salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 9471–9481.

[53] T. Zhou, D.-P. Fan, M.-M. Cheng, J. Shen, and L. Shao, "RGB-D salient object detection: A survey," *Computational Visual Media*, pp. 1–33, 2021.

[54] J. Han, H. Chen, N. Liu, C. Yan, and X. Li, "CNNs-based RGB-D saliency detection via cross-view transfer and multiview fusion," *IEEE Trans. Cybernetics*, vol. 48, no. 11, pp. 3171–3183, 2017.

[55] N. Wang and X. Gong, "Adaptive fusion for RGB-D salient object detection," *IEEE Access*, vol. 7, pp. 55 277–55 284, 2019.

[56] N. Huang, Y. Yang, D. Zhang, Q. Zhang, and J. Han, "Employing bilinear fusion and saliency prior information for RGB-D salient object detection," *IEEE Trans. Multimedia*, 2021.

[57] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, "MnasNet: Platform-aware neural architecture search for mobile," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 2820–2828.

[58] S. Mehta, M. Rastegari, L. Shapiro, and H. Hajishirzi, "ESPNetv2: A light-weight, power efficient, and general purpose convolutional neural network," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 9190–9200.

[59] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 10 781–10 790.

[60] Y. Liu, Y.-C. Gu, X.-Y. Zhang, W. Wang, and M.-M. Cheng, "Lightweight salient object detection via hierarchical visual perception learning," *IEEE Trans. Cybernetics*, vol. 51, no. 9, pp. 4439–4449, 2021.

[61] Y. Liu, X.-Y. Zhang, J.-W. Bian, L. Zhang, and M.-M. Cheng, "SAM-Net: Stereoscopically attentive multi-scale network for lightweight salient object detection," *IEEE Trans. Image Process.*, vol. 30, pp. 3804–3814, 2021.

[62] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[63] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Int. Conf. Mach. Learn.* Madison, WI, USA: Omnipress, 2010, pp. 807–814.

[64] Y.-H. Wu, S.-H. Gao, J. Mei, J. Xu, D.-P. Fan, R.-G. Zhang, and M.-M. Cheng, "JCS: An explainable covid-19 diagnosis system by joint classification and segmentation," *IEEE Trans. Image Process.*, vol. 30, pp. 3113–3126, 2021.

[65] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004.

[66] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in *2016 fourth international conference on 3D vision (3DV)*. IEEE, 2016, pp. 565–571.

[67] R. Cong, J. Lei, C. Zhang, Q. Huang, X. Cao, and C. Hou, "Saliency detection for stereoscopic images based on depth confidence analysis and multiple cues fusion," *IEEE Signal Processing Letters*, vol. 23, no. 6, pp. 819–823, 2016.

[68] H. Chen and Y. Li, "Progressively complementary-aware fusion network for RGB-D salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 3051–3060.

[69] Z. Zhang, Z. Lin, J. Xu, W.-D. Jin, S.-P. Lu, and D.-P. Fan, "Bilateral attention network for rgb-d salient object detection," *IEEE Trans. Image Process.*, vol. 30, pp. 1949–1961, 2021.

[70] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Annu. Conf. Neur. Inform. Process. Syst.* Curran Associates, Inc., 2019, pp. 8026–8037.

[71] S.-M. Hu, D. Liang, G.-Y. Yang, G.-W. Yang, and W.-Y. Zhou, "Jittor: a novel deep learning framework with meta-operators and unified graph execution," *Science China Information Sciences*, vol. 63, no. 12, pp. 1–21, 2020.

[72] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Int. Conf. Learn. Represent.*, 2015.

[73] Y. Niu, Y. Geng, X. Li, and F. Liu, "Leveraging stereopsis for saliency analysis," in *IEEE Conf. Comput. Vis. Pattern Recog.* IEEE, 2012, pp. 454–461.

[74] C. Zhu and G. Li, "A three-pathway psychobiological framework of salient object detection using stereoscopic technology," in *Int. Conf. Comput. Vis. Worksh.*, 2017, pp. 3008–3014.

[75] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 1597–1604.

[76] D.-P. Fan, M.-M. Cheng, Y. Liu, T. Li, and A. Borji, "Structure-measure: A new way to evaluate foreground maps," in *Int. Conf. Comput. Vis.*, 2017, pp. 4548–4557.

[77] D.-P. Fan, G.-P. Ji, X. Qin, and M.-M. Cheng, "Cognitive vision inspired object segmentation metric and loss function," *SCIENTIA SINICA Informationis*, vol. 6, 2021.

[78] L.-Z. Chen, Z. Lin, Z. Wang, Y.-L. Yang, and M.-M. Cheng, "Spatial information guided convolution for real-time RGBD semantic

segmentation," *IEEE Trans. Image Process.*, vol. 30, pp. 2313–2324, 2021.

[79] X. Chen, K.-Y. Lin, J. Wang, W. Wu, C. Qian, H. Li, and G. Zeng, "Bi-directional cross-modality feature propagation with separation-and-aggregation gate for RGB-D semantic segmentation," in *Eur. Conf. Comput. Vis.*, 2020, pp. 561–577.

[80] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Eur. Conf. Comput. Vis.*, 2012, pp. 746–760.

Yu-Chao Gu received his bachelor's degree from the Beijing University of Chemical Technology in 2019. He is currently pursuing the master's degree with the College of Computer Science, Nankai University. His research interests include efficient deep-learning and computer vision.

Yu-Huan Wu is currently a Ph.D. candidate with College of Computer Science at Nankai University, supervised by Prof. Ming-Ming Cheng. He received his bachelor's degree from Xidian University in 2018. His research interests include computer vision and machine learning.

Ming-Ming Cheng received his PhD degree from Tsinghua University in 2012. Then he did two years research fellow with Prof. Philip Torr in Oxford. He is now a professor at Nankai University, leading the Media Computing Lab. His research interests include computer graphics, computer vision, and image processing. He received research awards, including ACM China Rising Star Award, IBM Global SUR Award, and CCF-Intel Young Faculty Researcher Program. He is on the editorial boards of IEEE TPAMI and IEEE TIP.
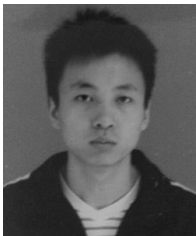
Yun Liu received his bachelor's and doctoral degrees from Nankai University in 2016 and 2020, respectively. Currently, he works as a postdoctoral scholar with Prof. Luc Van Gool at ETH Zurich. His research interests include computer vision and machine learning.

Jun Xu received his B.Sc. and M.Sc. degrees from School of Mathematics Science, Nankai University, Tianjin, China, in 2011 and 2014, respectively, and the Ph.D. degree from the Department of Computing, Hong Kong Polytechnic University, in 2018. He worked as a Research Scientist at IIAI, Abu Dhabi, UAE. He is currently a Lecturer with School of Statistics and Data Science, Nankai University. More information can be found at https://csjunxu.github.io/.

Jia-Wang Bian Jia-Wang Bian received the B.Eng. degree from Nankai University, where he was advised by Prof. M.-M. Cheng. He is currently pursuing the Ph.D. degree with The University of Adelaide. He was a Research Assistant with the Singapore University of Technology and Design (SUTD). He also did a Trainee Engineer Job with the Advanced Digital Sciences Center (ADSC), Huawei Technologies Co., Ltd., and Tusimple. He is also an Associated Ph.D. Researcher with the Australian Centre for Robotic Vision (ACRV). He is advised by Prof. I. Reid and Prof. C. Shen. His research interests include computer vision and robotics.