Depth Quality-Inspired Feature Manipulation for Efficient RGB-D Salient Object Detection

Wenbo Zhang  Ge-Peng Ji  Zhuo Wang
College of CS, Sichuan University  Inception Institute of AI  College of CS, Sichuan University
zhangwenbo@stu.scu.edu.cn  gepengai.ji@gmail.com  imwangzhuo@stu.scu.edu.cn

Keren Fu*  Qijun Zhao
College of CS, Sichuan University  College of CS, Sichuan University
fkrsuper@scu.edu.cn  qjzhao@scu.edu.cn

ABSTRACT

RGB-D salient object detection (SOD) recently has attracted increasing research interest by benefiting conventional RGB SOD with extra depth information. However, existing RGB-D SOD models often fail to perform well in terms of both efficiency and accuracy, which hinders their potential applications on mobile devices and real-world problems. An underlying challenge is that the model accuracy usually degrades when the model is simplified to have few parameters. To tackle this dilemma and also inspired by the fact that depth quality is a key factor influencing the accuracy, we propose a novel depth quality-inspired feature manipulation (DQFM) process, which is efficient itself and can serve as a gating mechanism for filtering depth features to greatly boost the accuracy. DQFM resorts to the alignment of low-level RGB and depth features, as well as holistic attention of the depth stream to explicitly control and enhance cross-modal fusion. We embed DQFM to obtain an efficient light-weight model called DFM-Net, where we also design a tailored depth backbone and a two-stage decoder for further efficiency consideration. Extensive experimental results demonstrate that our DFM-Net achieves state-of-the-art accuracy when comparing to existing non-efficient models, and meanwhile runs at 140ms on CPU (2.2x faster than the prior fastest efficient model) with only ~8.5Mb model size (14.9% of the prior lightest). Our code will be available at https://github.com/zwbx/DFM-Net.

KEYWORDS

RGB-D salient detection, neural networks, deep learning

1 INTRODUCTION

Salient object detection (SOD) aims to locate image regions that attract master human visual attention. It is useful in many downstream tasks, e.g., object segmentation [50], medical segmentation [15, 28], tracking [58], image/video compression [22]. Owing to the powerful representation ability of deep learning, great progresses have been made in SOD in recent years, but most of them use only RGB images as input to detect salient objects [16, 30, 49]. This is unavoidable to encounter challenges in complex scenarios, such as cluttered or low-contrast background.

With the popularity of depth sensors/devices, RGB-D SOD has become a hot research topic [18, 20, 39, 53, 55, 60], because additional useful spatial information embedded in depth maps could serve as a complementary cue for more robust detection [64]. Meanwhile, depth data is already widely available on many mobile devices [17], e.g., modern smartphones like Huawei Mate 40 Pro, iPhone 12 Pro, Samsung Galaxy S20+. This has opened up a new range of applications for the RGB-D SOD task. Unfortunately, the time-space consumption of existing approaches [6, 11, 17, 19, 32, 42, 54, 57, 62] is still too high, hindering their further applications on mobile devices and real-world problems. Therefore, an efficient and accurate RGB-D SOD model is highly desirable.

Motivation. We notice that unstable quality of depth is one key factor which largely influences the accuracy, as mentioned in [6, 8, 17]. However, very few existing models explicitly take this issue into consideration. We also argue that depth quality is difficult to determine solely according to a depth map [6, 8], because it is tough to judge whether a salient region in the depth map belongs

![Figure 1: Average probability distributions (solid curves) of edge Dice coefficients computed from “Good quality”, “Bad quality”, and “Mismatch” subsets (10 times random sampling), as well as the probability distributions (dash curves) of the whole STERE and SIP datasets. The right images show three examples from the three subsets, respectively.](image-url)
to noise or a target object, as exemplified in Fig. 1 (b). Since RGB-D SOD concerns two paired images as input, i.e., an RGB image and a depth map, our observation is that a high-quality depth map usually has some boundaries well-aligned to the corresponding RGB image. We call this observation “boundary alignment” (BA). To validate such a BA observation, we randomly choose 50 paired samples (tagged as “Good quality”) as shown in Fig. 1 from SIP [17] dataset, and also 50 “Bad quality” samples from STERE [38] dataset. Choosing such two datasets is based on the general observations of previous works [17, 19]. Additionally, we construct a new set of samples from the “Good quality” set, tagged as “Mismatch”, by randomly mismatching the RGB and depth images of the “Good quality” set, to see if this behavior can be reflected by BA. Note that this behavior actually causes no changes to individual RGB or depth images, therefore having no impact to a depth quality measurement (e.g., [8]) that is dependant only on depth itself (often called no-reference metric [37]). To determine boundary alignment, an off-the-shelf edge detector [23] is used to obtain two edge maps from RGB and depth, respectively. We calculate their Dice coefficients [36] as a measure of BA. The probability distributions (average of 10 times random sampling) of Dice coefficients are shown in Fig. 1, where the three sets of samples correspond to different colors. We can see that BA seems a strong evidence for the depth quality issue, and meanwhile for the “Mismatch” set, its Dice coefficients are generally lower than those of the “Good quality” set.

Inspired by the above fact that the alignment of low-level RGB and depth features can somewhat reflect depth quality, we propose a new depth quality-inspired feature manipulation (DQFM) process. The intuition behind DQFM is to assign lower weights to depth features if the quality of depth is bad, effectively avoiding injecting noisy or misleading depth features to improve detection accuracy for efficient models. We also augment DQFM with depth holistic attention, in order to enhance depth features when the depth quality is judged to be good. With the help of DQFM, we explicitly control and enhance the role of depth features during cross-modal fusion. Further in this paper, we embed DQFM into an encoder-decoder framework to obtain an efficient light-weight model called DFM-Net (Depth Feature Manipulation Network), where a tailored depth backbone and a two-stage decoder are designed for efficiency consideration. The main contributions are summarized as follows:

- We propose an efficient depth quality-inspired feature manipulation (DQFM) process, to explicitly control and enhance depth features during cross-modal fusion. DQFM avoids injecting noisy or misleading depth features, and can effectively improve detection accuracy for efficient models.
- Benefited from DQFM, we propose an efficient light-weight model DFM-Net (Depth Feature Manipulation Network), which has a tailored depth backbone and a two-stage decoder.
- Compared to 15 state-of-the-art (SOTA) models, DFM-Net is able to achieve superior accuracy, meanwhile running at 7 FPS on CPU (2.2× faster than the prior fastest model) with only ~8.5Mb model size (14.9% of the prior smallest).

2 RELATED WORK

The utilization of RGB-D data for SOD has been extensively explored for years. Based on the goal of this paper, in this section, we review general RGB-D SOD methods, as well as previous works on efficient models and depth quality analyses.

2.1 General RGB-D SOD Methods

Traditional methods mainly rely on hand-crafted features [7, 9, 21, 45]. Lately, deep learning-based methods have made great progress and gradually become a mainstream [4, 10, 17–19, 29, 32, 39, 42, 54, 57, 62, 63, 65]. Qu et al. [44] first introduced CNNs to infer object saliency from RGB-D data. Zhu et al. [65] designed a master network to process RGB data, together with a sub-network for depth data, and then incorporated depth features into the master network. Fu et al. [19] utilized a Siamese network for simultaneous RGB and depth feature extraction, which discovers the commonality between these two views from a model-based perspective. Zhang et al. [54] proposed a probabilistic network via conditional variational auto-encoders to model human annotation uncertainty. Zhang et al. [57] proposed a complementary interaction fusion framework to locate salient objects with fine edge details. Liu et al. [11] introduced a selective self-mutual attention mechanism that can fuse attention learned from both modalities. Li et al. [33] designed a cross-modal weighting network to encourage cross-modal and cross-scale information fusion from low-, middle- and high-level features. Fan et al. [18] adopted a bifurcated backbone strategy to split multi-level features into student and teacher ones, in order to suppress distractors within low-level layers. Pang et al. [39] provided a new perspective to utilize depth information, in which the depth and RGB features are combined to generate region-aware dynamic filters to guide the decoding in the RGB stream. Li et al. [32] proposed a cross-modality feature modulation module that enhances feature representations by taking depth features as prior. Luo et al. [35] utilized graph-based techniques to design a network architecture for RGB-D SOD. Ji et al. [29] proposed a novel collaborative learning framework, where multiple supervision signals are employed, yielding a depth-free inference method. Zhao et al. [63] designed a single stream network to directly take a depth map as the fourth channel of an RGB image, and proposed a depth-enhanced dual attention module. A relatively complete survey on RGB-D SOD can be found in [64].

Despite that encouraging detection accuracy has been obtained by the above RGB-D SOD methods, most of them have heavy models and are computationally expensive.

2.2 Efficient RGB-D SOD Methods

Besides the above-mentioned methods, several recent methods attempt to take model efficiency into consideration. Specific techniques are used to reduce high computation brought by multi-modal feature extraction and fusion. Piao et al. [43] employed knowledge distillation for a depth distiller, which aims at transferring depth knowledge obtained from the depth stream to the RGB stream, thus allowing a depth-free inference framework. Chen et al. [12] constructed a tailored depth backbone to extract complementary features. Such a backbone is much more compact and efficient than classic backbones, e.g., ResNet [24] and VGGNet [48]. Besides, the method adopts a coarse-to-fine prediction strategy that simplifies

---

1The concept of “efficient model” is hard to define. Here and also in the following experiment section, we consider that efficient models should be less than 100Mb.
the top-down refinement process. The utilized refinement module is in a recurrent manner, further reducing model parameters.

2.3 Depth Quality Analyses in RGB-D SOD

Since the quality of depth often affects model performance, a few researchers have considered the depth quality issue in RGB-D SOD, so as to alleviate the impact of low-quality depth. As early attempts, some works proposed to conduct depth quality assessment from a global perspective and obtained a quality score. Cong et al. [8] first proposed a no-reference depth quality metric [37] to alleviate the contamination of low-quality depth. Later, Fan et al. [17] used two networks sharing the same structure to process RGB-D input and depth input individually. Depth quality is then evaluated by comparing between the results from the two networks. Cong et al. [6] proposed to predict the depth quality score via a perceptron with high-level RGB and depth features as input. Such a perceptron was trained by scores calculated by comparing thresholded depth maps with ground truth.

Instead of a single quality score, more recently, spatial quality evaluation of depth maps was also considered, in order to find valuable depth region. Wang et al. [51] designed three hand-crafted features to excavate depth following multi-scale methodology. Chen et al. [3] proposed to locate the “Most Valuable Depth Regions” of depth by comparing pseudo GT generated from a sub-network with RGB-D as input, with two saliency maps generated from two sub-networks with RGB/depth as input.

Different from the above existing methods that have high time-space complexity, our depth quality assessment is much more efficient and is more suitable to benefit a light-weight model. Besides, our quality module is end-to-end trainable and is also unsupervised ([3, 6] are supervised for quality estimation).

3 METHODOLOGY

3.1 Overview

Fig. 2 shows the block diagram of the proposed DFM-Net, which consists of encoder and decoder parts. For efficiency consideration, our encoder part follows the design in [18], where the RGB branch is simultaneously responsible for RGB feature extraction and cross-modal fusion between RGB and depth features. The decoder part, on the other hand, conducts simple two-stage fusion to generate the final saliency map. More specifically, the encoder consists of an RGB-related branch which is based on MobileNet-v2 [24], a depth-related branch which is a tailored efficient backbone, and also the proposed DQFM. Both branches lead to five feature hierarchies, and the output stride is 2 for each hierarchy except 1 for the last hierarchy. The extracted depth features at a certain hierarchy, after passing through a DQFM gate, are then fused into the RGB branch by simple element-wise addition before being sent to the next hierarchy. Besides, in order to capture multi-scale semantic information, we add a PPM (pyramid pooling module [61]) at the end of the RGB branch. Note that in practice, the DQFM gate contains two successive operations, namely depth quality-inspired weighting (DQW) and depth holistic attention (DHA).

Let the features from the five RGB/depth hierarchies be denoted as \( f^r_i \) \((m \in \{r, d\}, i = 1, ..., 5)\), the fused features be denoted as \( f^d_i \) \((i = 1, ..., 5)\), and the features from PPM be denoted as \( f^p_i \). The aforementioned cross-modal feature fusion can be expressed as:

\[
\begin{align*}
\text{DQW}^{c} &= f^d_i = f^d_i \odot (\alpha_i \odot \beta_i \odot f^d_i), \\
\text{DHA} &= f^d_i \odot (\alpha_i \odot \beta_i \odot f^d_i),
\end{align*}
\]

where \(\alpha_i\) and \(\beta_i\) are computed by DQW and DHA to manipulate depth features \(f^d_i\) to be fused, and \(\odot\) denotes element-wise multiplication.

After encoding as shown in Fig. 2, \(f^d_i\) \((i = 1, ..., 5)\) and \(f^p_i\) are fed to the subsequent decoder part.

\(^{2}\text{If the multipliers’ dimensions are different, before element-wise multiplication, the one with less dimension will be replicated to have the same dimension of the other(s).}\)
3.2 Depth Quality-Inspired Feature Manipulation (DQFM)

DQFM consists of two key components, namely DQW (depth quality-inspired weighting) and DHA (depth holistic attention), which generate $\alpha_i$ and $\beta_i$ in Eq. 1, respectively. $\alpha_i \in \mathbb{R}^i$ is a scalar determining “how much” depth features are involved, while $\beta_i \in \mathbb{R}^{k \times k}$ ($k$ is the feature size at hierarchy $i$) is a spatial attention map, deciding “what regions” to focus in the depth features. Below we describe the internal structures of DQW and DHA.

**Depth Quality-Inspired Weighting (DQW).** Inspired by the aforementioned BA observation in Sec. 1, as shown in Fig. 3, DQW learns the weighting term $\alpha_i$ adaptively from low-level features $f_r^1$ and $f_d^1$, because such low-level features characterize image edges/boundaries [59]. To this end, we first apply convolutions to $f_r^1/f_d^1$ to obtain transferred features $f_r'/f_d'$, which are expected to capture more edge-related activation:

$$f_r' = \text{BConv}_{1 \times 1}(f_r^1), \quad f_d' = \text{BConv}_{1 \times 1}(f_d^1),$$

where $\text{BConv}_{1 \times 1}(-)$ denotes a $1 \times 1$ convolution with BatchNorm and ReLU activation. To evaluate low-level feature alignment and inspired by Dice coefficient [36], given edge activation $f_r'$ and $f_d'$, the alignment feature vector $V_{BA}$ that encodes the alignment between $f_r'$ and $f_d'$ is computed as:

$$V_{BA} = \frac{\text{GAP}(f_r' \otimes f_d')}{\text{GAP}(f_r' + f_d')},$$

where $\text{GAP}(-)$ denotes the global average pooling operation, and $\otimes$ means element-wise multiplication. To make $V_{BA}$ robust against slight edge shifting, we propose to compute $V_{BA}$ at multi-scale and concatenate the results to generate an enhanced vector. As shown in Fig. 3, such multi-scale calculation is conducted by subsequently downsampling the initial features $f_r'/f_d'$ by max pooling with stride 2, and then computing $V_{BA}^1/V_{BA}^2$ of the same as Eq. 3. Suppose $V_{BA}^1$, $V_{BA}^2$, and $V_{BA}^{mss}$ are the alignment feature vectors computed as three scales as shown in Fig. 3, the enhanced vector $V_{BA}^{ms}$ is computed as:

$$V_{BA}^{ms} = [V_{BA}^1, V_{BA}^2, V_{BA}^{ms}],$$

where $[\cdot]$ denotes channel concatenation. Next, two fully connected layers are applied to derive $\alpha \in \mathbb{R}^2$ from $V_{BA}^{ms}$:

$$\alpha = \text{MLP}(V_{BA}^{ms}).$$

where $\text{MLP}(-)$ denotes a two-layer perceptron with the Sigmoid function at the end. Thus the vector $\alpha$ contains entries $\alpha_i \in (0, 1) \ (i = 1, 2, ..., 5)$ as its elements. Notably, here we adopt different weighting terms for different hierarchies rather than an identical one. The effectiveness of this strategy is validated in Sec. 4.4.

**Depth Holistic Attention (DHA).** Depth holistic attention (DHA) enhances depth features spatially, by deriving holistic attention map $\beta_i$ from the depth stream. Technically as in Fig. 4, we first utilize the highest-level features $f_d^5$ from the depth stream to locate coarse salient regions (with supervision signals imposed as shown in Fig. 2). To facilitate subsequent pixel-wise operations, we compress and then up-sample $f_d^5$ into $f_{dht}$, which has the same dimension as $f_r'/f_d'$, formulated as:

$$f_{dht} = F_{UP}(\text{BConv}_{1 \times 1}(f_d^5)).$$

where $F_{UP}$ means 8x bilinear up-sampling. Then we combine low-level RGB and depth features to recalibrate $f_{dht}$. Similar to the computation of $V_{BA}$, we first transfer $f_r'/f_d'$ to $f_r'/f_d'$ as in DQW. The resulting features are element-wisely multiplied to generate features $f_{ec}$, which emphasizes common edge-related activation. To better model long-range dependencies across low-level and high-level features while maintaining efficiency for DHA, we employ the max pooling operation and dilated convolution to rapidly increase receptive fields. The recalibration process is defined as:

$$F_{rec}(f_{dht}) = F_{UP} \left( \text{DConv}_{3 \times 3}(F_{DN}(f_{dht} + f_{ec})) \right),$$

where $F_{rec}(-)$ denotes once recalibration process. $\text{DConv}_{3 \times 3}(-)$ denotes the $3 \times 3$ dilated convolution with stride 1 and dilation rate 2, followed by BatchNorm and ReLU activation. $F_{UP}(-)/F_{DN}(-)$ denotes bilinear up-sampling/down-sampling operation to $2/(\frac{1}{2})$ times the original size. As a trade-off between performance and
Table 1: Detailed structure of the proposed tailored depth backbone (TDB), which is based on inverted residual bottleneck blocks (IRB) of MobileNet-V2 [47]. About notations, $t$: expansion factor of IRB, $c$: output channels, $n$: times the block is repeated, and $s$: stride of hierarchy, which is applied to the first block of the repeating blocks.

| Input | Output | Block | $t$ | $c$ | $n$ | $s$ |
|-------|--------|-------|-----|-----|-----|-----|
| 256 $\times$ 256 $\times$ 1 | 128 $\times$ 128 $\times$ 16 | IRB | 3 | 16 | 1 | 2 |
| 128 $\times$ 128 $\times$ 16 | 64 $\times$ 64 $\times$ 24 | IRB | 3 | 24 | 3 | 2 |
| 64 $\times$ 64 $\times$ 24 | 32 $\times$ 32 $\times$ 32 | IRB | 3 | 32 | 7 | 2 |
| 32 $\times$ 32 $\times$ 32 | 16 $\times$ 16 $\times$ 96 | IRB | 2 | 96 | 3 | 2 |
| 16 $\times$ 16 $\times$ 96 | 16 $\times$ 16 $\times$ 320 | IRB | 2 | 320 | 1 | 1 |

We first use $3 \times 3$ depth-wise separable convolution with BatchNorm and ReLU activation, denoted as $\text{DSConv}_{3 \times 3}$, to compress the encoder features ($f_{\ell}^i$, $i = 1, 2, ..., 6$) to a unified channel 16, denoted as “CP” in Fig. 2. Then we use the well-known channel attention operator [27] $F_{CA}$ to enhance features by weighting different channels, denoted as “CA” in Fig. 2. The above process can be described as:

$$c_{\ell}^i = F_{CA}(\text{DSConv}_{3 \times 3}(f_{\ell}^i)), \quad \text{(11)}$$

where $c_{\ell}^i$ denotes the compressed and enhanced features. To reduce feature hierarchies, inspired by [18], we group 6 hierarchies into 3.

3.3 Tailored Depth Backbone (TDB)

Usually, depth is less informative than RGB. Hence, we consider using a tailored depth backbone (TDB), which is lighter, as a trade-off between efficiency and accuracy. Specifically, we base our TDB on the inverted residual bottleneck blocks (IRB) from MobileNet-V2 [47], and construct a new smaller backbone with reduced channel numbers and stacked blocks, whose structure is detailed in Tab. 1. As a result, our TDB is much lighter than previous light-weight backbones, e.g., (Ours: only 0.9Mb, ATSA’s [56]: 6.7Mb, PGAR’s [12]: 6.2Mb, MobileNet-V2: 6.9Mb), and meanwhile, its performance is slightly better than MobileNet-V2 (see Sec. 4.4). During training, we embed TDB into DFM-Net without pre-training, and supervision signals are imposed at the end of the backbone to enforce saliency feature learning from depth, as shown in Fig. 2. The coarse prediction result obtained from TDB is formulated as:

$$S_d = F_p^d(f_d^i)). \quad \text{(10)}$$

where $S_d$ means the coarse prediction from TDB, which is supervised by ground truth (GT). $F_p^d(·)$ denotes a prediction head consisting of a $1 \times 1$ convolution followed by a BatchNorm layer and Sigmoid activation, and also 16× bilinear up-sampling to recover the original input size. The effectiveness of the proposed TDB will be validated in Sec. 4.4.

3.4 Two-Stage Decoder

Unlike the popular U-Net [46] which adopts the hierarchy-by-hierarchy top-down decoding strategy, we propose a simplified two-stage decoder, including pre-fusion and full-fusion, to further improve efficiency. The pre-fusion aims to reduce feature channels and hierarchies, by channel compression and hierarchy grouping, denoted as “CP” and “G” in Fig. 2. Based on the outputs of pre-fusion, the full-fusion further aggregates low-level and high-level hierarchies to generate the final saliency map.

Pre-fusion Stage. We first use $3 \times 3$ depth-wise separable convolution with BatchNorm and ReLU activation, denoted as $\text{DSConv}_{3 \times 3}$, to compress the encoder features ($f_{\ell}^i$, $i = 1, 2, ..., 6$) to a unified channel 16, denoted as “CP” in Fig. 2. Then we use the well-known channel attention operator [27] $F_{CA}$ to enhance features by weighting different channels, denoted as “CA” in Fig. 2. The above process can be described as:

$$c_{\ell}^i = F_{CA}(\text{DSConv}_{3 \times 3}(f_{\ell}^i)), \quad \text{(11)}$$

Instead of the common $\text{BConv}$, depth-wise separable convolution $\text{DSConv}$ is used here for large numbers of input channels.
where $c_f^{\text{low}}$ is bilinear up-sampling to $i$ times the original size.

**Full-fusion Stage.** Since in the pre-fusion stage, the channel numbers and hierarchies are already reduced, in the full-fusion stage, we directly concatenate high-level and low-level hierarchies, and then feed the concatenation to a prediction head to achieve the final full-resolution prediction map, denoted as:

$$S_C = F_p\left(c_f^{\text{low}}, F_{\text{up}}(c_f^{\text{high}})\right),$$

where $S_C$ is the final saliency map, and $F_p(\cdot)$ indicates a prediction head consisting of two $3 \times 3$ depth-wise separable convolutions (followed by BatchNorm layers and ReLU activation), a $3 \times 3$ convolution with Sigmoid activation, as well as a $2 \times$ bilinear up-sampling to recover the original input size.

### 3.5 Loss Function

The overall loss $L$ is composed of the final loss $L_C$ and deep supervision for the depth branch loss $L_d$, formulated as:

$$L = L_C(S_C, G) + L_d(S_d, G),$$

where $G$ denotes the ground truth (Similar to previous losses [17–19, 39], we use the standard cross-entropy loss for $L_C$ and $L_d$.}

---

**Table 2: Quantitative benchmark results.** $\gamma$ for a metric denotes that a larger/smaller value is better. Our results are highlighted in bold. The scores/numbers better than ours are underlined (efficient and non-efficient models are labeled separately).

| Metric | PCF | CYPR | MMPC | PR | CYPR | ECPP | DMRB | DIINet | IL-DFC | UCNet | SIF | S2MA | CoNet | cmMS | ECMSS | DANet | ATSA | DFS-Net† | Ours | ASD | PGAR | DFS-Net† |
|--------|-----|------|------|----|------|------|------|--------|--------|--------|-----|------|-------|------|-------|-------|------|---------|-----|-----|-----|---------|
| CPU (ms) | 35762 | 46886 | 1694 | 521 | 7858 | 1192 | 520 | 119 | 125 | 330 | 167 | 430 | 102 | 123 | 93 | 57 | 62 | 8.5 |
| GPU (FPS) | 64 | 64 | 64 | 64 | 64 | 64 | 64 | 64 | 64 | 64 | 64 | 64 | 64 | 64 | 64 | 64 | 64 | 64 | 64 |

---

**4 EXPERIMENTS AND RESULTS**

#### 4.1 Datasets and Metrics

We conduct experiments on six widely used datasets, including NJU2K [31] (1,985 samples), NLPR [41] (1,000 samples), STERE [38] (1000 samples), RGBD135 [7] (135 samples), LFSD [34] (100 samples), and SIP [17] (929 samples). Following previous works [19, 54, 62], we use the same 1,500 samples from NJU2K and 700 samples from NLPR for training, and test on the remaining samples. Four commonly used metrics are used for evaluation, including S-metric ($S_C$) [5], maximum F-measure ($F^{\text{max}}$) [1], maximum E-measure ($E^{\text{max}}$) [13, 14], and mean absolute error (MAE, $M_A$) [2]. For efficiency analysis, we report model size (Mb, Megasize), inference time (ms, millisecond) on CPU, and FPS (frame-per-second) on GPU.

#### 4.2 Implementation Details

Experiments are conducted on a workstation with Intel Core i7-8700 CPU, Nvidia GTX 1080Ti GPU with CUDA 10.1. We implement DFS-Net by PyTorch [40], and RGB and depth images are both resized to 256 $\times$ 256 for input. For testing, the inference time on CPU and FPS on GPU is obtained by averaging 100 times inference with batch size 1, and no any data augmentation or post-processing is used. For training, in order to balance the network on limited training samples, following [18], we apply various data augmentation techniques i.e., random translation/cropping, horizontal flipping, color enhancement and so on. We train DFS-Net for 300 epochs on a single 1080Ti GPU, taking about 4 hours. The initial learning rate is set as $1e-4$ for Adam optimizer, and the batch size is 10. The poly learning rate policy is used, where the power is set to 0.9.
Table 3: Ablation analyses for DQFM, where the effectiveness of DQW and DHA is validated. Details are in Sec. 4.4.

| #DQW | DHA | Size (MB) | NLPR [41] | NUI2K [31] | RGBD135 [7] | LSDF [34] | STERE [38] |
|------|-----|-----------|-----------|-----------|-------------|-----------|-----------|
| 1    | ✓   | 8.499     | 0.884    | 0.897     | 0.079       | 0.890     | 0.887     |
| 2    | ✓   | 8.416     | 0.878    | 0.922     | 0.054       | 0.898     | 0.902     |
| 3    | ✓   | 8.451     | 0.861    | 0.908     | 0.063       | 0.891     | 0.901     |
| 4    | ✓   | 8.545     | 0.883    | 0.926     | 0.051       | 0.923     | 0.908     |

Table 4: Effectiveness of the recalibration process $F_{rec}$ in DHA. The number below “$F_{rec}$” in the table means the times of recalibration. Specifically, zero means that no recalibration is conducted.

| $F_{rec}$ | $\alpha$ | $\beta$ | $\gamma$ | $\delta$ | $\epsilon$ | $\zeta$ |
|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| 5        | 0         | 0.877     | 0.881    | 0.919     | 0.054     | 0.890     |
| 6        | 1         | 0.572     | 0.873    | 0.918     | 0.056     | 0.920     |
| 7        | 2         | 0.883     | 0.887    | 0.926     | 0.051     | 0.923     |
| 8        | 3         | 0.864     | 0.868    | 0.916     | 0.059     | 0.921     |

Figure 7: Performance visualization. The vertical axis indicates the accuracy ($F_{max}$) on SIP [17]. The horizontal axis indicates the CPU speed (FPS). The circle size is proportional to the model size. More details refer to Tab. 2.

4.3 Quantitative and Qualitative Comparisons

Since we cannot expect that an extremely light-weight model can always outperform existing non-efficient models (which are much larger), we have also extended DFM-Net to obtain a larger but more powerful model called DFM-Net*, by replacing MobileNet-v2 used in the RGB branch with ResNet-34 [24]. Then to align the features from the RGB branch to those from TDB, slight modifications are made. Results of DFM-Net and DFM-Net*, compared to 15 state-of-the-art (SOTA) models including PCF [10], MMCI [4], CPFP [62], DRMA [42], D3Net [17], JL-DCF [19], UCNet [54], SSF [57], S2MA [11], CoNet [29], cmMS [32], DANet [63], ATSA [56], A2dele [43] and PGAR [12], can be found in Tab. 2.

As shown in Tab. 2, DFM-Net surpasses existing efficient models A2dele [43] and PGAR [12] significantly on detection accuracy and model size, as well as CPU speed. It runs at 140ms on CPU, which is the fastest among all the contenders, with only ~8.5Mb model size (14.9% of that of A2dele). About GPU speed, DFM-Net ranks the second after A2dele [43]. This is because the depth-wise separable convolution layers, which have been extensively used in MobileNet-v2 and TDB, cannot be fully accelerated on GPU in the current implementation [25, 52]. On the other hand, one can see that DFM-Net* achieves SOTA performance when compared to existing non-efficient models, with absolutely the fastest CPU/GPU speed and the smallest model size. Visual comparisons with representative methods are shown in Fig. 6, where our results are closer to the ground truth (GT).

To better reflect the advantages of the proposed method, as shown in Fig. 7, we visualize all models by plotting their accuracy ($F_{max}$ on SIP dataset) and model size (proportional to the circle diameter). It is clearly seen that DFM-Net and DFM-Net* can rank the most upper right with very small circles, indicating that our method can perform well in terms of both efficiency and accuracy when compared to existing techniques.

4.4 Ablation Studies

We conduct thorough ablation studies on six datasets by removing or replacing components from the full implementation of DFM-Net.
Effectiveness of DQFM. DQFM consists of two key components, namely DQW and DHA. Tab. #3 shows different configurations by ablating DQW/DHA. In detail, #1 denotes a baseline model which has removed both DQW and DHA from DFM-Net. Configuration #2 and #3 mean having either one component, while #4 means the full model of DFM-Net. Basically, from Tab. 3 one can see that incorporating either DQW and DHA into the baseline model #1 leads to consistent improvement on almost all datasets. Besides, comparing #2/#3 to #4, we see that employing both DQW and DHA can further enhance the results, demonstrating the complementary effect between DQW and DHA. The underlying reason could be that, although DHA is able to enhance potential target regions in the depth, it is unavoidable to make some mistakes (e.g., highlight wrong regions) especially in low depth quality cases. Luckily, DQW somewhat relieves such a side-effect because in the meantime it assigns low global weights to depth features. In all, these two components can work cooperatively to improve the robustness of the network, as we mentioned in Sec. 3.2. Last but not least, from the model sizes of different configurations shown in Tab. 3, the extra model parameters for introducing DQW and DHA are quite few. This rightly meets our goal and implies that they could potentially become universal components for light-weight models in the future.

In Fig. 8, we show visual examples of setting #3 (namely without DQW) and #4. We also visualize the magnitudes of $\alpha$, namely the mean value $\overline{\alpha}$ of $\alpha_1 \sim \alpha_5$. From Fig. 8 (a) and (b), we can see that incorporating DQW does help improve detection accuracy, and practically, DQW is able to function as expected, namely rendering the good quality case with higher weights ($\overline{\alpha} = 0.65$), and vice versa ($\overline{\alpha} = 0.29$). In the good quality case (a), it is difficult to distinguish between the shadow and the man’s legs in the RGB view, but this can be done easily in the depth view. Incorporating DQW to give more emphasis to depth features, therefore, can help better separate the entire human body from the shadow. In the bad quality case (b), although in the depth view the cup handle is much blurry, the impact of misleading depth has been alleviated, and the whole object still can be detected out accurately.

Recalibration in DHA. As described in Sec. 3.2, in DHA, we utilize operation $F_{rec}$ to recalibrate the coarse information from high-level depth features. To validate the necessity of using $F_{rec}$, we experiment with different times (from 0 to 3) of using $F_{rec}$-wise. These variants are denoted as #5, #6, #4, and #7 in Tab. 4. Note that #4 corresponds to the default implementation of DFM-Net. From Tab. 4, we can see that #4 (recalibrate twice) achieves the overall best performance. The underlying reason should be that, appropriate usage of $F_{rec}$ can expand the coverage areas of attention maps to make conservative filtering for object edges as well as some inaccurately located objects, but too large receptive field leads to over-dilated attention regions that are less informative. This is the reason why when the times increase to 3, the performance starts to degenerate on most datasets, except on STERE whose depth quality is generally low, which easily leads to inaccurate attention location.

DQFM Gating Strategy. As we mentioned in Sec. 3.2, we adopt a multi-variable strategy for $\alpha_1$ and $\beta_i$. To validate this strategy, we compare it to the single-variable strategy, namely using identical (only one) $\alpha_i$ and $\beta_i$. Tab. 5 shows the results, from which it can be seen that our proposed multi-variable strategy is better, because it somewhat increases the network flexibility by rendering different hierarchies with different quality weights and attention maps.

Tailored Depth Backbone. The effectiveness of TDB is validated by comparing it to MobileNet-V2. We implement a configuration #9 by switching TDB directly to MobileNet-V2, while maintaining all other settings unchanged. Evaluation results are shown in Tab. 6. We can see that our tailored depth backbone is more efficient (0.9Mb vs. MobileNet-V2’s 6.9Mb) and also more accurate (with noticeable gains) than MobileNet-V2 when utilized in DFM-Net. This demonstrates the feasibility of using a lighter backbone to process depth data for efficiency purpose.

5 CONCLUSION

In this paper, we propose an efficient RGB-D SOD model called DFM-Net, characterized by the the DQFM process to explicitly control and enhance depth features during cross-modal fusion. The two key components in DQFM, namely DQW and DHA, are validated by comprehensive ablation experiments. The experimental results show that DQW and DHA are both essential for obtaining higher detection accuracy with very few model parameters added on. Besides, a tailored depth backbone and a two-stage decoder are elaborately designed to further improve the efficiency of DFM-Net. Our DFM-Net achieves new state-of-the-art records on light-weight model size as well as CPU speed, meanwhile retaining decent accuracy. In the future, it is very attractive to apply DFM-Net on some embedding or mobile systems that process RGB-D data.
REFERENCES

[1] R. Achanta, S. Hemami, F. J. Estrada, and S. Süssstrunk. 2009. Frequency-tuned salient region detection. In CVPR 1597–1604.

[2] A. Borji, M. Cheng, Huaizhu Jiang, and J. Li. 2015. Salient Object Detection: A Benchmark. IEEE TPAMI 37 (2015) 5706–5722.

[3] C. Chen, J. Wei, C. Peng, and H. Qin. 2021. Depth-Quality-Aware Salient Object Detection. IEEE Transactions on Image Processing PP, 99 (2021), 1–1.

[4] Hao Chen, Youfu Li, and Dan Su. 2019. Multi-modal fusion network with multi-scale locality-path and cross-modal interactions for RGB-D salient object detection. Pattern Recognition 86 (2019), 376–385.

[5] Ming-Ming Chen and Deng-Ping Fan. 2021. Cognitive Depth-Quality-Inspired Feature Manipulation for Efficient RGB-D Salient Object Detection Conference’21, October 2021, Chengdu, China.

[6] Yuan Cong, Jianjun Lei, Changqing Zhang, Q. Huang, Xiaochun Cao, and C. Hou. 2016. Saliency Detection for Stereoscopic Images Based on Depth Confidence Analysis and Multiple Cues Fusion. IEEE Signal Processing Letters 23 (2016), 819–823.

[7] Cipadi et al. 2013. An in depth view of saliency. In BMVC. Georgia Institute of Technology, 112.

[8] H. Chen et al. 2018. Progressively Complementarily-Aware Fusion Network for RGB-D Salient Object Detection. In CVPR 3951–3960.

[9] Nian Liu et al. 2020. Learning Selective Self-Mutual Attention for RGB-D Saliency Detection. In CVPR 3675–3684.

[10] Shuhan Chen et al. 2020. Rethinking RGB-D salient object detection: Models, datasets, and large-scale evidence analysis. In CVPR.

[11] N. Liu et al. 2020. Enhanced-alignment Measure for Binary Foreground Map Evaluation. In ICIP.

[12] Deng-Ping Fan, Ge-Peng Ji, Xuebin Qin, and Ming-Ming Cheng. 2021. Cognitive Vision Inspired Object Segmentation Metric and Loss Function (in Chinese). SCIENTIA SINICA Informationis (2021). https://doi.org/10.1360/SS2020-0370

[13] Deng-Ping Fan, Ge-Peng Ji, Tao Zhou, Geng Chen, Huaizhu Fu, Jianbing Shen, and Ling Shao. 2020. PraNet: Parallel Reverse Attention Network for Polyp Segmentation. MICCAI (2020).

[14] Deng-Ping Fan, Tengpeng Li, Zheng Lin, Chenlei Guo, and Liming Zhang. 2021. A Novel Multiresolution Spatiotemporal Saliency Detection Model and Its Applications in Image and Video Compression. IEEE TIP 19, 1 (2020), 185–198.

[15] Jianzhong He, Shiliang Zhang, Ming Yang, Yanhu Shan, and Tiejun Huang. 2019. Potentiality-Aware Gated Attention Network for RGB-D Salient Object Detection. In CVPR.

[16] Runmin Cong, Jianjun Lei, Changqing Zhang, Q. Huang, Xiaochun Cao, and C. Hou. 2020. Asymmetric Two-Stream Architecture for Accurate RGB-D Saliency Detection. In CVPR.

[17] Yao Jiang, Tao Zhou, Ge-Peng Ji, Keren Fu, Qijun Zhao, and Deng-Ping Fan. 2020. Field Light Salient Object Detection: A Review and Benchmark. arXiv preprint arXiv:2010.04968 (2020).

[18] Ran Ju, L. Ge, W. Geng, T. Ren, and Gangshan Wu. 2014. Depth saliency based on anisotropic center-surround difference. In IJCB 1115–1119.

[19] Jing Zhang, Deng-Ping Fan, Yuchao Dai, Saeed Anwar, Fatemeh Saleh, Tingting Yang, and Nick Barnes. 2020. UC-Net: Uncertainty Informed RGB-D Saliency Detection via Conditional Variational Autoencoders. In CVPR 8582–8591.

[20] Jing Zhang, Deng-Ping Fan, Yuchao Dai, Sadegh Alikhahi, and Nick Barnes. 2021. Uncertainty Inspired RGB-D Saliency Detection. IEEE TPAMI (2021). https://doi.org/10.1109/TPAMI.2021.3073564

[21] Shuhan Chen et al. 2020. Rethinking RGB-D salient object detection: Models, datasets, and large-scale evidence analysis. In CVPR.

[22] Liucun Yang, Huaizhu Fu, and Jianbing Shen. 2021. Re-thinking co-salient object Segmentation. arXiv:2010.04968 (2020).

[23] Yongri Piao, Wei Ji, Jingjing Li, Miao Zhang, and Huchuan Lu. 2019. Depth-induced multi-scale recurrent attention network for saliency detection. In ECCV 7254–7263.

[24] Yongri Piao, Zhengkun Rong, Miao Zhang, Weisong Ren, and Huchuan Lu. 2020. Azdele: Adaptive and Attentive Depth Distiller for Efficient RGB-D Salient Object Detection. In CVPR.

[25] Youwei Pang, Lihe Zhang, Xiaogao Zhao, and Huchuan Lu. 2020. Hierarchical Dynamic Filtering Network for RGB-D Saliency Object Detection. In CVPR 235–252.

[26] Pingping Zhang, Wei Liu, Dong Wang, Yinjie Lei, and Huchuan Lu. 2019. Non-Local Attention U-Net: Convolutional Networks for Volumetric Medical Image Segmentation. 3DV (2019), 565–571.

[27] Shuhan Chen et al. 2020. Rethinking RGB-D salient object detection: Models, datasets, and large-scale evidence analysis. In CVPR.

[28] AiMish Mittal, Anush K. Moorthy, and A. Bovik. 2012. No-Reference Image Quality Assessment in the Spatial Domain. IEEE TIP 21 (2012), 4695–4708.

[29] Youwei Pang, Lihe Zhang, Xiaogao Zhao, and Huchuan Lu. 2020. Hierarchical Dynamic Filtering Network for RGB-D Saliency Object Detection. In CVPR 235–252.

[30] Runmin Cong, Jianjun Lei, Changqing Zhang, Q. Huang, Xiaochun Cao, and C. Hou. 2020. Asymmetric Two-Stream Architecture for Accurate RGB-D Saliency Detection. In CVPR.

[31] Yao Jiang, Tao Zhou, Ge-Peng Ji, Keren Fu, Qijun Zhao, and Deng-Ping Fan. 2020. Field Light Salient Object Detection: A Review and Benchmark. arXiv preprint arXiv:2010.04968 (2020).

[32] Ran Ju, L. Ge, W. Geng, T. Ren, and Gangshan Wu. 2014. Depth saliency based on anisotropic center-surround difference. In IJCB 1115–1119.

[33] Chongyi Li, Runmin Cong, Piao Yongyi, Xu Qianquan, and Chen Changye. 2020. RGB-D salient object detection with cross-modality modulation and selection. In ECCV 225–241.

[34] Qiongyang Li, Zhouzhuo Liu, L. Ye, Yang Wang, and H. Ling. 2020. Cross-Modal Weighting Network for RGB-D Saliency Object Detection. In ECCV 665–681.

[35] Nianli Li, J. Ye, Y. Ji, Habin Ling, and J. Yu. 2014. Saliency Detection on Light Field. IEEE TPAMI 39 (2017), 1605–1616.

[36] Yao Jiang, Tao Zhou, Ge-Peng Ji, Keren Fu, Qijun Zhao, and Deng-Ping Fan. 2020. Field Light Salient Object Detection: A Review and Benchmark. arXiv preprint arXiv:2010.04968 (2020).

[37] cups.com Conference’21, October 2021, Chengdu, China.
[59] Pingping Zhang, D. Wang, H. Lu, Hongyu Wang, and X. Ruan. 2017. Amulet: Aggregating Multi-level Convolutional Features for Salient Object Detection. In ICCV. 202–211.

[60] Zhao Zhang, Zheng Lin, Jun Xu, Wen-Da Jin, Shao-Ping Lu, and Deng-Ping Fan. 2021. Bilateral attention network for RGB-D salient object detection. IEEE TIP 30 (2021), 1949–1961.

[61] Hengshuang Zhao, J. Shi, Xiaojuan Qi, Xiaogang Wang, and J. Jia. 2017. Pyramid Scene Parsing Network. In CVPR. 6230–6239.

[62] Jia-Xing Zhao, Yang Cao, Deng-Ping Fan, Ming-Ming Cheng, Xuan-Yi Li, and Le Zhang. 2019. Contrast prior and fluid pyramid integration for RGBD salient object detection. In CVPR. 3927–3936.

[63] Xiaoqi Zhao, Lihe Zhang, Youwei Pang, Huchuan Lu, and Lei Zhang. 2020. A Single Stream Network for Robust and Real-time RGB-D Salient Object Detection. In ECCV. 520–538.

[64] Tao Zhou, Deng-Ping Fan, Ming-Ming Cheng, Jianbing Shen, and Ling Shao. 2020. RGB-D Salient Object Detection: A Survey. Computational Visual Media 7 (2020), 37–69.

[65] Chunbiao Zhu, Xing Cai, Kan Huang, Thomas H Li, and Ge Li. 2019. PDNet: Prior-model guided depth-enhanced network for salient object detection. In ICME. 199–204.