Multi-Guidance CNNs for Salient Object Detection

SHUAIXIONG HUI, QIANG GUO, and XIAOYU GENG, School of Computer Science and Technology, Shandong University of Finance and Economics, and Shandong Provincial Key Laboratory of Digital Media Technology, China
CAIMING ZHANG, School of Software, Shandong University, China

Feature refinement and feature fusion are two key steps in convolutional neural networks–based salient object detection (SOD). In this article, we investigate how to utilize multiple guidance mechanisms to better refine and fuse extracted multi-level features and propose a novel multi-guidance SOD model dubbed as MGuid-Net. Since boundary information is beneficial for locating and sharpening salient objects, edge features are utilized in our network together with saliency features for SOD. Specifically, a self-guidance module is applied to multi-level saliency features and edge features, respectively, which aims to gradually guide the refinement of lower-level features by higher-level features. After that, a cross-guidance module is devised to mutually refine saliency features and edge features via the complementarity between them. Moreover, to better integrate refined multi-level features, we also present an accumulative guidance module, which exploits multiple high-level features to guide the fusion of different features in a hierarchical manner. Finally, a pixelwise contrast loss function is adopted as an implicit guidance to help our network retain more details in salient objects. Extensive experiments on five benchmark datasets demonstrate our model can identify salient regions of an image more effectively compared to most of state-of-the-art models.

CCS Concepts: • **Computing methodologies → Interest point and salient region detections;**

Additional Key Words and Phrases: Salient object detection, self-guidance, cross-guidance, multi-level feature aggregation, pixelwise contrast loss

ACM Reference format:
Shuaixiong Hui, Qiang Guo, Xiaoyu Geng, and Caiming Zhang. 2023. Multi-Guidance CNNs for Salient Object Detection. *ACM Trans. Multimedia Comput. Commun. Appl.* 19, 3, Article 117 (February 2023), 19 pages.
https://doi.org/10.1145/3570507

This work was supported in part by the National Natural Science Foundation of China under Grant 61873145, in part by the Natural Science Foundation of Shandong Province for Excellent Young Scholars under Grant ZR2017JL029, and in part by the Science and Technology Innovation Program for Distinguished Young Scholars of Shandong Province Higher Education Institutions under Grant 2019KJN045.
Authors' addresses: S. Hui, Q. Guo (corresponding author), X. Geng, School of Computer Science and Technology, Shandong University of Finance and Economics, and Shandong Provincial Key Laboratory of Digital Media Technology, No. 7366, East Erhuan Road, Jinan, China, 250014; emails: huishuaixiong@mail.sdufe.edu.cn, guoqiang@sdufe.edu.cn, gengxiaoyu@mail.sdufe.edu.cn; C. Zhang, School of Software, Shandong University, No. 1500, Shunhua Road, Jinan, China, 250101; email: czhang@sdu.edu.cn.

1551-6857/2023/02-ART117
https://doi.org/10.1145/3570507

1 INTRODUCTION

As a preprocessing procedure of various computer vision tasks, **salient object detection (SOD)**, which aims to acquire the most visually prominent regions in images, has received increasing attention in the past decades [4, 16, 32, 35, 36]. Benefiting from the ability of **convolutional neural networks (CNNs)** to extract both the high-level semantic information and the low-level spatial details, deep CNN-based SOD models have triggered a significant breakthrough [2, 14, 23, 58].

Two key issues of CNN-based SOD are how to acquire high-quality saliency features and how to better integrate these features. To address the first issue, many feature refinement strategies were constructed, such as embedding saliency prior knowledge [2, 38, 51], introducing edge detection stream [12, 46, 56], using recurrent refinement modules [15, 30, 39], and so on. Among them, the strategies based on edge detection stream have attracted more attention due to the complementarity between binary segmentation map and salient edge map. For example, in Reference [12] the edge features produced by edge detection stream are aggregated into the saliency features of corresponding level to refine the boundaries of salient objects. Unlike Reference [12], which only utilizes a unidirectional refinement framework from edge features to saliency features, SCRN [46] refines saliency features and edge features simultaneously through stacking multiple bidirectional refinement modules. To handle the second issue, various fusion strategies were explored to fully utilize each level feature [14, 22, 24, 25, 34, 43, 45, 53, 60]. A representative strategy is Amulet [53], which integrates the extracted multi-level features into different resolutions and then adaptively combines them for generating final saliency maps. Besides, another popular fusion strategy is DSS [14]. It aggregates higher-level features into lower-level features by short connections, enabling each fused features with both the semantics and the spatial details.

Different from the aforementioned feature refinement and fusion methods, this article investigates how to employ the internal relationships between different features to build various guidance mechanisms, so as to better refine and aggregate extracted multi-level features. As boundary information can help to segment salient objects from background regions, two parallel decoders are employed to generate multi-level saliency features and edge features simultaneously. To refine these extracted features, a self-guidance module is devised and applied to multi-level saliency and edge features, respectively. It is capable of focusing the information of foreground regions and enhancing the adaptability of the network in different scenarios. Additionally, a cross-guidance module is used between saliency features and edge features, taking full advantage of their complementarity. To integrate refined features better, an accumulative guidance module is constructed, which can gradually aggregate multi-level features via hierarchical structure. And in this module, a fusion-enhanced unit is designed to further enlarge the receptive field of network and alleviate the aliasing effect of upsampling. In addition, a new pixelwise contrast loss function is adopted, which acts as an implicit guidance, to help our network capture more precise saliency features.

To summarize, the contributions of this article are as follows:

- To better refine the extracted multi-level features, we build a self-guidance module and a cross-guidance module. The former can highlight the salient regions in each level feature and suppress the distractors in low-level features to some extent, while the latter can mutually refine saliency features and edge features and make boundary information more accurate.
- To aggregate the refined features more fully, we construct an accumulative guidance module that takes full advantage of the high-level semantic information and low-level spatial details. Moreover, one fusion-enhanced unit is designed so that each spatial position in the fused features can view the semantic information of different scale spaces.
- To enhance the robustness of our network and further improve the accuracy of salient objects, we also define a new pixelwise contrast loss function that can implicitly guide the predicted saliency map to approach the ground truth map at pixel level.

- Based on the aforementioned guidance mechanisms, a novel multi-guidance SOD model is proposed, termed MGuid-Net.

In addition, we conduct extensive experiments on five benchmark datasets, and the results demonstrate that the proposed MGuid-Net yields competitive performance compared with 10 state-of-the-art methods. At the same time, comprehensive ablation experiments are performed to validate the effectiveness of each component in our model.

2 RELATED WORK

In this section, we briefly review some representative guidance strategies, multi-level feature refinement methods, and multi-level feature fusion methods used in deep SOD architectures. We refer readers to the recent and comprehensive survey paper [40] for more details.

2.1 Guidance Strategy

Various guidance strategies have been presented to assist deep network in detecting salient objects better within existing SOD models. In Reference [38], the hand-crafted prior knowledge (i.e., color prior, intensity prior, orientation prior, and central prior) is encoded into the saliency prior map to guide the training process of the whole network, which can make the network focus more on the foreground regions of an image. Although such guidance strategy can boost the performance of SOD to a certain extent, the calculation of prior knowledge greatly decreases the inference speed of network. Therefore, some methods try to embed guidance mechanism into the network structure, resulting in an end-to-end trainable model. For instance, a progressive attention guided network was devised in Reference [54] to selectively aggregate contextual information from different level features. It uses a guidance mechanism as a part of the network to connect the two attention features of adjacent stages. At the same time, benefiting from the powerful feature extraction ability of fully convolutional neural networks [29], an alternative guidance strategy between different features was explored in Reference [56]. It utilizes salient edge features as a guidance to refine salient object features at different scales, in which the edge features are generated by integrating global location information and local boundary information. Unlike the above two methods, ASNet [42] handles both salient object detection and human fixation prediction tasks simultaneously. As a result, the saliency features are progressively optimized under the guidance of the fixation map in a coarse-to-fine and top-down manner. Moreover, ASNet is equipped with several loss functions derived from current widely adopted evaluation metrics for implicitly guiding the network to learn more representative features.

Additionally, the guidance mechanisms also play an important role in weakly supervised SOD and RGB-D SOD. Li et al. [20] established a guidance strategy directly on the attention maps to force the weakly supervised inference network to focus more on the whole of salient objects. In Reference [59] a pre-training model based on an RGB benchmark dataset was exploited to guide the learning of RGB-D master network for addressing the lack of annotated RGB-D data.

2.2 Multi-level Feature Refinement

The methods of using refinement strategies to improve the quality of extracted multi-level features have achieved great success in deep SOD. A typical strategy is employing edge features as an auxiliary information to refine multi-level saliency features. In Reference [44], the saliency and edge features are simultaneously refined by sharing feature between the mask and boundary sub-networks. Besides, BCNet [7] first uses multiple consistency saliency maximization modules to produce the global edge feature and then incorporates it into the saliency features of each level to refine them. Since attention mechanisms are able to selectively retain effective information through the change of weight, it is also widely used in feature refinement. AFNet [11] adopts a

series of attentive feedback modules to indirectly refine the extracted features of each level by controlling the message passing between encoders and decoders. In Reference [43], PAGE-Net uses a stacked attention to refine multi-level saliency features. It can enhance the representation ability of different network layers and further expand the receptive field of the network. Beyond the methods above, various recurrent modules are used to refine the extracted features iteratively. Kuen et al. [17] built a **recurrent attentional network (RACDNN)** that learns context information from the previous timestep to refine each level feature in the next timestep. Different from RACDNN, RADF [15] alternatively refines both the features of each layer and the multi-level integrated features in a recurrent way. Note that the integrated features are generated by upsampling and concatenating the features with different scales. In Reference [41], another general recurrent model with deeper supervision strategy was proposed to gradually refine the extracted multi-level features. It simulates the interactive processes of human perception and alternately deploys the top-down and bottom-up saliency inference in an iterative and cooperative manner.

2.3 Multi-level Feature Fusion

Due to the use of downsampling and pooling operations in deep SOD models, the resolution of extracted features is gradually reduced with the increase of network depth. Therefore, the deeper layer features can generally capture more semantic information that contributes to locate the specific positions of salient objects, while abundant spatial details are retained in the shallower layer features. To accurately obtain salient objects with sharp boundaries, it is necessary to integrate multi-level features better.

The ELD-Net [18] directly aggregates both the high-level semantic feature and the encoded low-level distance feature into a fully connected neural network classifier for evaluating the saliency of each image region. However, in ELD-Net, the intermediate hierarchical features with rich information are not taken into account. To this end, a hierarchical recurrent convolutional neural network was built in Reference [26], which integrates each level feature into the high-level semantic feature step by step in a top-down manner. It is worth noting that both recurrent convolutional and upsampling operations are used to process the fused feature in each step. Similarly, BDMPM [52] also adopts a top-down fusion strategy, but the difference is that it employs a bidirectional framework with gate function to selectively propagate information between multi-level features before fusing them. Although the aforementioned two methods show their capability to enhance the quality of saliency map, directly integrating low-level features into high-level features will introduce some additional interference (i.e., the details in non-salient regions) into fused features. And the global location information acquired from deeper layer features will be weakened during the progressive fusion. To alleviate this problem, the PoolNet architecture [24] directly transfers high-level semantic information to each lower-level features and employs a feature aggregation module in the top-down fusion process. This makes PoolNet provide the location information of potential salient objects for different level features as well as screen out some distractors in low-level features.

3 PROPOSED METHOD

3.1 Overall Architecture

The overall architecture of MGuid-Net is illustrated in Figure 1, which mainly contains the following components except backbone network (Resnet50 [13]): **self-guidance module (SGM)**, **cross-guidance module (CGM)**, and **accumulative guidance module (AGM)**. To be specific, two SGMs are adopted to gradually deliver the high-level semantic information to low-level spatial features via layerwise guidance. After that, a CGM is used to bidirectionally pass messages between saliency features and edge features, which takes the outputs of SGMs as inputs. Finally,

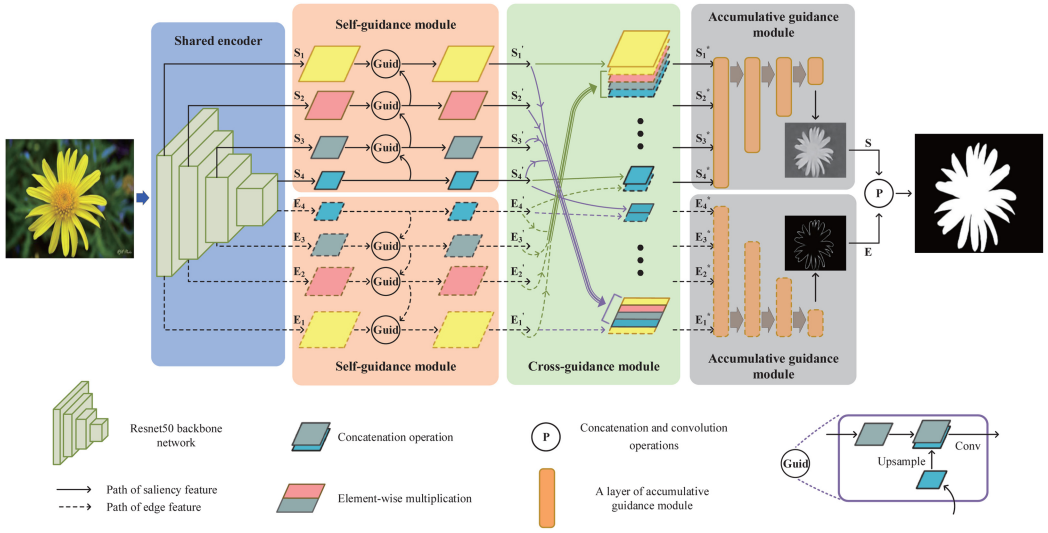


Fig. 1. The overall architecture of the proposed MGuid-Net. The parallelograms with solid and dotted borders denote the saliency features (S) and edge features (E), respectively. Different level features are highlighted in different colors. In cross-guidance module, the thick lines are used as pipelines to simultaneously transmit multi-level features.

two AGMs are established to aggregate refined saliency features and edge features, respectively. The final saliency map is generated by performing simple concatenation and convolution operations on the aggregated saliency feature and edge feature. In addition, MGuid-Net is equipped with a novel **pixelwise contrast loss (PCL)** function except for the binary cross entropy loss to enhance the robustness of our network. To facilitate description, the extracted saliency features from low-level to high-level are denoted as S_1, S_2, S_3, and S_4, respectively. Similarly, the multi-level edge features are represented as E_1, E_2, E_3, and E_4.

3.2 Self-Guidance Module

In the extracted multi-level features, high-level features with semantic information can coarsely localize the foreground regions, while low-level features with rich details are utilized to sharpen the boundary regions of salient objects. Therefore, to make our network pay more attention to the foreground regions and enhance its adaptive ability in different scenarios, a SGM is built upon multi-level saliency features and edge features, respectively. It can increase the weights of foreground regions and reduce ones of background regions by progressively transferring high-level semantic information to low-level features.

For saliency features, SGM starts from the high-level feature S_4 to transfer semantics layer by layer into lower-level features. As shown in Figure 1, S_4 is first upsampled to the same size as S_3, and then S_3 and S_4 are concatenated to generate the refined saliency feature $S_3^{'}$ through a convolution operation. With this, we use $S_3^{'}$ and S_2 to produce $S_2^{'}$ by the concatenation and convolution operations. Finally, the refined $S_1^{'}$ can be also obtained in the same way. Mathematically, the SGM for saliency features can be formulated as

$$S_i^{'} = \begin{cases} Conv\left(Cat\left(Up(S_{i+1}^{'}), S_i\right)\right), & i = 1, 2, 3 \\ S_i, & i = 4, \end{cases} \tag{1}$$

where $S_i^{'}$ is the refined saliency feature through SGM; $Conv$ and Cat are convolution and concatenation operations, respectively; and Up represents upsampling operation with a factor of 2. Similarly, the SGM for edge features can be given as

$$E_i^{'} = \begin{cases} Conv\left(Cat\left(Up(E_{i+1}^{'}), E_i\right)\right), & i = 1, 2, 3 \\ E_i, & i = 4, \end{cases} \tag{2}$$

where $E_i^{'}$ is the refined edge feature through SGM.

3.3 Cross-guidance Module

To take full advantage of edge information, we further explore the interrelations between saliency features and edge features. For the SOD task, saliency features should emphasize the whole salient objects, while edge features emphasize the boundary information of salient objects. Therefore, the logical relationships between saliency and edge features in SOD can be represented as $S \cap E = E$ and $S \cup E = S$, where S and E denote saliency features and edge features, respectively. Based on the logical relationships, a CGM is proposed, which can not only guide the refinement of saliency features by edge features but also utilize saliency features to filter out the boundary information of non-salient regions in edge features. Note that using lower-level features to guide higher-level features will add more distractors, i.e., spatial details in background regions. Thus, when CGM refines a certain level saliency feature, the lower level edge features will not be used. Similarly, such operations also exist in the refinement process of edge features. For example, the saliency feature $S_2^{'}$ is refined by edge features $E_2^{'}, E_3^{'}, E_4^{'}$ to generate a new saliency feature S_2^{*}, but the highest-level saliency feature $S_4^{'}$ is only refined by the edge feature $E_4^{'}$. Technically speaking, since it is challenging to directly apply intersection ($\cap$) and union ($\cup$) operations in feature level, we use the multiplication and concatenation operators to approximate these two operations in logical relationships, respectively. Thus, in CGM, the whole refinement process of saliency features can be denoted as

$$S_i^{*} = S_i^{'} + Conv\left(\left(Cat_{j=i}^{4}\left(Up(E_j^{'}), S_i^{'}\right)\right)\right), \quad i = 1, 2, 3, 4, \tag{3}$$

and the refinement process of edge features is formulated as follows:

$$E_i^{*} = E_i^{'} + Conv\left(\prod_{j=i}^{4}\left(Up(S_j^{'}), E_i^{'}\right)\right), \quad i = 1, 2, 3, 4, \tag{4}$$

where S_i^{*} and E_i^{*} are saliency feature and edge feature refined by CGM, respectively; Up is upsampling operation with scale-factor equal to 2^{j-i}; and $\prod$ means pixelwise multiplication.

It is worth noting that the concatenation and multiplication operators in Equations (3) and (4) cannot be substituted for each other from a logical perspective. The reasons are twofold: First, in the refinement process of edge features, if the multiplication operator is replaced by the concatenation one, the distractors in the non-salient regions cannot be filtered out, although the boundary information of the salient regions can be emphasized to a certain extent. Second, when saliency features are refined, the use of multiplication operator will lead to the loss of some information in the salient regions, degrading the performance of SOD. Technically, although the modified formulas can still work, their performance will be degraded. To verify the above analyses and demonstrate the effectiveness of the strategy used in this article, we specially conduct four comparison experiments on the DUTS benchmark dataset, and their quantitative results are shown in Table 1. We can find that the strategy used in this article (i.e., No. 1) achieves the best performance. Although the other three combinations also work, we believe that this is due to the residual connection in Equations (3) and (4) rather than the correct use of concatenation and multiplication operators.

Table 1. Quantitative Comparisons of Different Strategies Used in Equations (3) and (4)

No.	Strategy		DUTS-TE	
	Equation (3)	Equation (4)	MAE↓	maxF↑
1	concatenation	multiplication	**0.037**	**0.869**
2	multiplication	concatenation	0.047	0.848
3	concatenation	concatenation	0.041	0.867
4	multiplication	multiplication	0.040	0.866

No. 1 represents the strategy used in this article.

3.4 Accumulative Guidance Module

To better integrate the refined multi-level features from CGM, an AGM with hierarchical structure is introduced into MGuid-Net. It makes full use of high-level semantics and low-level spatial details through stacking and reusing each level feature in a hierarchical manner so that the fused features contain more complete saliency information. Meanwhile, the fused high-level features can also guide the refinement of low-level features more accurately, resulting in more precise salient object boundaries. Besides, a **fusion-enhanced unit (FEU)** is embedded into AGM to further deal with these features after fusion.

Figure 2 illustrates the detailed structure of AGM, which adopts a hierarchical structure with four layers. The nth layer ($n = 1, 2, 3, 4$) contains $(5 - n)$ different level features, and each level feature is integrated with all higher-level features to generate one new feature for the next layer. Take AGM for saliency features as an example, the inputs of first layer are the refined features from CGM, i.e., $S_1^*, S_2^*, S_3^*, S_4^*$. Then S_2^*, S_3^*, and S_4^* are upsampled to the size of S_1^* and fed into the FEU together with S_1^* to produce the lowest-level feature of second layer. Similarly, the mid-level feature of the second layer can be obtained by performing FEU on both the S_2^* and upsampled S_3^*, S_4^*. In this way, AGM can acquire each level feature in the second, third, and fourth layers.

Note that the upsampling operation based on bilinear interpolation can break the semantic information of high-level features and cause the aliasing effect of upsampling, when the upsampling rate is too large (e.g., 8). To address this issue, we use the FEU to process fused features. As shown in the top-right corner of Figure 2, a fused feature is first mapped to different scale spaces by different pooling strides (1, 2, 4, 8) to independently extract features from each scale space. After that, these extracted features are merged together and sent to a convolution layer to generate the enhanced feature. Although the upsampling operation with a factor of 8 is also used in FEU, other scale features extracted from the same fused feature can act as supplementary information to gradually make up the missing semantic information and alleviate the aliasing effect of upsampling. Moreover, FEU further expands the receptive field of our network and enables the enhanced features contain the semantic information of different scales in each spatial location.

3.5 Pixelwise Contrast Loss

As pointed out in References [3, 9], the contrast between foreground and background regions and the uniform distributions in these two regions are crucial for SOD. Intuitively, this prior information can be effectively utilized by designing a specific loss function. Previous work [55] employs a **contrast loss (CL)** to implicitly guide the contrast-enhanced net for improving the quality of depth maps captured from state-of-the-art sensors, which proves that it is feasible to enhance the contrast through a loss function. Specifically, the CL includes three items: the foreground object distribution loss, the background distribution loss, and the whole depth image distribution loss. The first two items aim to make the internal distributions of foreground and background regions more uniform.

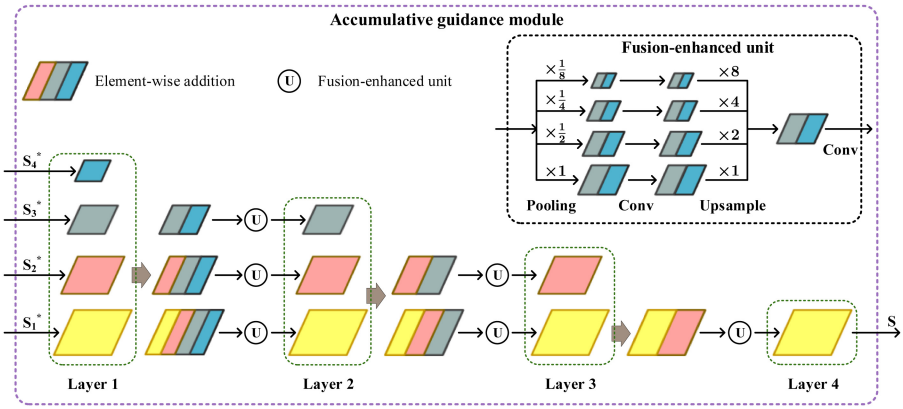


Fig. 2. Detailed structure of accumulative guidance module. The $\times 1, \times \frac{1}{2}, \ldots, \times 4, \times 8$ represent the change of feature size in fusion-enhanced unit.

The third item can enhance the contrast between these two regions. Inspired by this, we design a new PCL function, which can enlarge the contrast between foreground and background regions while the internal distributions of these two regions become uniform. Unlike CL that contains three hyper-parameters, our PCL has only one hyper-parameter (α defined in Equation (8)). Combining the PCL with the **binary cross entropy (BCE)** loss function, a hybrid loss function is finally generated to assist our network in capturing more details of salient objects and enhancing its robustness.

Here, the key idea of PCL is to make the pixel values in predicted saliency maps close to their ground truth as much as possible. Therefore, we define the d_F and d_B that respectively indicate the average difference of pixels in foreground and background regions between the predicted saliency maps and the ground truth maps. Besides, to reduce the amount of calculation, a sigmoid function is employed to compress the input of PCL to [0,1]. In detail, the d_F and d_B are computed as follows:

$$d_F = \sum_{i \in F} \frac{(X_i - \mathbf{1})^2}{n_F - 1}, \tag{5}$$

$$d_B = \sum_{i \in B} \frac{(X_i - \mathbf{0})^2}{n_B - 1}, \tag{6}$$

where F and B represent the regions corresponding to foreground and background in the ground truth map, respectively; X is the predicted saliency map; and n_F and n_B are the number of pixels in F and B. Obviously, according to the above definition, it can be concluded that both the d_F and d_B range from 0 to 1. Hence, to facilitate gradient descent, the log function with good derivability is utilized to expand the average difference (i.e., d_F and d_B), which is formulated as

$$\mathcal{L}_F = -log(1 - d_F), \quad \mathcal{L}_B = -log(1 - d_B). \tag{7}$$

Then, our PCL can be represented as

$$loss_{PCL} = \mathcal{L}_F + \alpha \mathcal{L}_B, \tag{8}$$

where α is a hyper-parameter to compromise the role of the two terms. And experimental results show that our MGuid-Net is insensitive to the selection of hyper-parameter α, see Section 4.3 for more details.

The BCE loss is a commonly used loss function in binary classification tasks. In deep SOD models, it usually measures the difference between predicted saliency maps and ground truth maps,

which is given as

$$loss_{BCE}(X, G) = -\sum_{i=1}^{w \times h} \Big[G_i \log (X_i) + (1 - G_i) \log (1 - X_i) \Big], \tag{9}$$

where G is the ground truth map and w and h denote the width and height of input image, respectively.

Finally, the hybrid loss function for training our model is defined as follows:

$$loss = loss_{PCL} + loss_{BCE}. \tag{10}$$

4 EXPERIMENTS

4.1 Implementation Details

Our MGuid-Net is implemented in Pytorch 1.8.1 with Python 3.6 and is run on a PC with 2.90-GHz CPU, 32G memory, and RTX 3060 GPU. The weights of backbone network are initialized by the pre-trained weights of Resnet50 on ImageNet [6]. And the parameters of other convolution layers are randomly initialized. We utilize the stochastic gradient descent to train our network in an end-to-end manner, in which momentum and weight decay are set as 0.9 and 0.0005, respectively. Moreover, the input images (original images and ground truth maps) are resized to 352×352. The maximum iteration step is set to 40 with a batch size of 7. The learning rate is initialized to 0.001 and divided by 10 at 20 epochs.

4.2 Datasets and Evaluation Metrics

Extensive experiments are conducted on five benchmark datasets, including DUTS (DUTS-TR and DUTS-TE) [37], DUT-OMRON [49], ECSSD [48], HKU-IS [19], and PASCAL-S [21]. Specifically, we use the benchmark dataset DUTS-TR to train our network and use DUT-OMRON, ECSSD, HKU-IS, PASCAL-S, and DUTS-TE for testing.

DUTS is a large-scale SOD benchmark dataset that contains 10,553 training images (DUTS-TR) and 5,019 testing images (DUTS-TE). DUTS-TR is collected from the ImageNet DET training/val sets [6], and DUTS-TE is gathered from both the ImageNet DET test set and the SUN dataset [47]. DUT-OMRON dataset with natural images consists of 5,168 images, and each image has at least one salient object and relatively complex background. The 1,000 images in ECSSD dataset are obtained from the internet, which are semantically meaningful and structurally complex. HKU-IS dataset contains 4,447 images, most of which have multiple salient objects and low contrast between salient object and backgrounds. PASCAL-S is currently a very challenging benchmark dataset, which is built on PASCAL VOC 2010 segmentation dataset [8]. In the PASCAL-S, 850 natural images with complex scenario is included. Of course, all the images in the datasets mentioned above have corresponding ground truth maps.

Four widely used metrics (i.e., **mean absolute error (MAE)** [5], **precision–recall (PR)** curve, **maximum F-measure (maxF)** [1], and **maximum E-measure (maxE)** [10]) are adopted to evaluate the performance of our MGuid-Net and other competing methods. The MAE is pixelwise mean absolute error between predicted saliency map (X) and ground truth map (G), which can be formulated as

$$MAE = \frac{1}{w \times h} \sum_{i=1}^{w \times h} |X_i - G_i|, \tag{11}$$

where w and h indicate the width and height of images and X_i and G_i refer to the pixel value of predicted saliency map and ground truth map at the pixel i, respectively. The PR curve consists of a series of value pairs of precision and recall, in which precision reflects the proportion of salient

Table 2. Ablation Analyses of the MGuid-Net and Quantitative Comparison of Different Modules

No.	Network Architecture	Visualization Results	SGM	CGM	AGM$^-$	AGM	PCL	DUTS-TE	
								MAE↓	maxF↑
1	Res50 (baseline)	Figure 3(j)						0.133	0.584
2	Res50+SGM	Figure 3(i)	√					0.045	0.839
3	Res50+CGM	Figure 3(f)		√				0.043	0.859
4	Res50+SGM+CGM	Figure 3(e)	√	√				0.041	0.861
5	Res50+AGM$^-$	Figure 3(h)			√			0.046	0.842
6	Res50+AGM	Figure 3(g)				√		0.044	0.853
7	MGuid-Net$^-$	Figure 3(d)	√	√		√		0.038	0.867
8	MGuid-Net	Figure 3(c)	√	√		√	√	0.037	0.869

Resnet50 is employed as a baseline and denoted as Res50. AGM$^-$ and MGuid-Net$^-$ represent the AGM without FEU and the MGuid-Net without PCL, respectively. In addition, to better display the compatibility between each module, the results of Res50+SGM+CGM, MGuid-Net$^-$, and MGuid-Net are marked in blue, green, and red, respectively.

pixels correctly detected in the predicted salient objects (X_O) while recall denotes the ratio of correctly detected salient pixels to foreground regions (G_O) of ground truth. Mathematically, the precision and recall can be computed as follows:

$$\text{Precision} = \frac{|X_O \bigcap G_O|}{|X_O|}, \quad \text{Recall} = \frac{|X_O \bigcap G_O|}{|G_O|}, \tag{12}$$

where $|\cdot|$ represents the number of pixel in one region. In contrast to PR curve, the F-measure metric considers both precision and recall, which can more comprehensively illustrate the quality of a saliency result. Its definition is as follows:

$$F_\beta = \frac{(1+\beta^2)\ \text{Precision} \times \text{Recall}}{\beta^2\ \text{Precision} + \text{Recall}}, \tag{13}$$

where β^2 is set to 0.3 [1]. Note that this article adopts the maxF that depicts the maximum value of F-measure under different thresholds. Besides, the maxE is also widely used to measure the quality of various SOD models [10, 31, 45], because it can simultaneously consider local and global matching degree between X and G.

4.3 Ablation Experiments

To validate the effectiveness of each component in our model, extensive ablation experiments are conducted on the DUTS benchmark dataset, and their quantitative and visual results are listed in Table 2 and Figure 3, respectively. For fair comparison, the basic network architecture and the settings of hyper-parameters are identical in all ablation experiments.

The effectiveness of SGM and CGM. To confirm the effectiveness of SGM and CGM in MGuid-Net and evaluate their respective contributions, we conduct three ablation experiments, i.e., Res50+SGM, Res50+CGM, and Res50+SGM+CGM. The experimental results in rows 1 and 2 of Table 2 show that the performance of Res50 (baseline) is greatly improved by introducing SGM. The MAE score is declined from 0.133 to 0.045, and the maxF is promoted from 0.584 to 0.839. This is because that SGM utilizes the different level features in the network compared with Res50 (baseline) and refines these features by gradually transmitting semantic information from high level to low level. Moreover, the introduction of CGM also improves the performance of baseline network to a large extent (see rows 1 and 3). But unlike the behavior of SGM, CGM pays more attention to edge information, which further refines saliency features and edge features through mutual

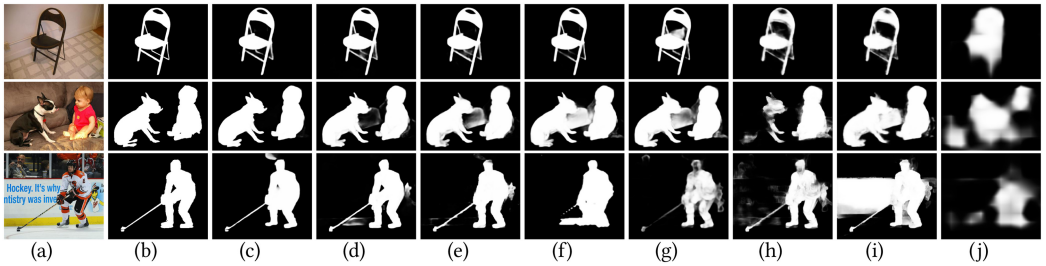


Fig. 3. The visualization results of ablation analyses, in which (a) shows original images, (b) shows the ground truth; and (c)–(j) correspond to the results of different network architectures: (c) MGuid-Net, (d) MGuid-Net$^-$, (e) Res50+SGM+CGM, (f) Res50+CGM, (g) Res50+AGM, (h) Res50+AGM$^-$, (i) Res50+SGM, and (j) Res50 (baseline).

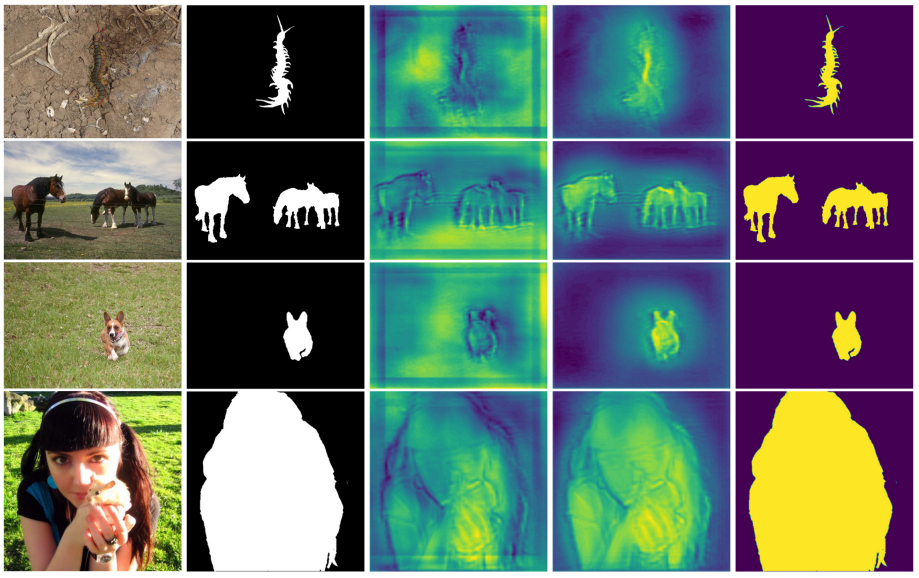
guidance between them. Meanwhile, the Res50+SGM+CGM is designed to verify the compatibility of SGM and CGM. As seen in rows 2–4 of Table 2, Res50+SGM+CGM outperforms Res50+SGM and Res50+CGM by 9.8% and 4.9% in terms of MAE score, 2.6% and 0.2% in terms of maxF. This means that SGM and CGM have good compatibility.

The visualization results in Figure 3(e), (f), (i), and (j) further validate our analyses above. The saliency maps of Res50+SGM contain more complete salient objects compared with that of Res50+CGM and Res50 (baseline), while more accurate salient object boundaries are presented in the results of Res50+CGM. And the saliency maps of Res50+SGM+CGM are obviously superior to the results of other three network architectures.

The effectiveness of AGM. The Res50+AGM$^-$ and Res50+AGM are performed to manifest the effectiveness of both the fusion strategy and FEU proposed in AGM. From rows 1, 5, and 6 of Table 2, it can be observed that Res50+AGM$^-$ and Res50+AGM achieve superior performances over the Res50 (baseline). Additionally, by comparing the results in rows 5 and 6, we can find that the introduction of FEU can obtain a performance gain of 1.3% in terms of maxF and decline of 4.5% in terms of MAE score. The main reason is that FEU can compensate for the loss of semantic information in the fused features caused by a too-large upsampling ratio. Meanwhile, the features before and after the FEU enhancement are visualized in Figure 4 for presenting the effectiveness of FEU more intuitively. Quite evidently, the features after enhancement ($\text{FEU}_{\text{after}}$) contain more accurate semantic information and less noise compared to ones before enhancement ($\text{FEU}_{\text{before}}$).

Besides, AGM is integrated into Res50+SGM+CGM to form the MGuid-Net$^-$ for proving the compatibility of AGM, SGM, and CGM. As shown in rows 4 and 7 of Table 2, MGuid-Net$^-$ can achieve significant performance gain, in contrast to Res50+SGM+CGM. The maxF is promoted from 0.861 to 0.867 and the MAE score is decreased from 0.041 to 0.038. The above results again indicate the effectiveness of AGM and its contribution to our network.

The effectiveness of PCL and the analysis of hyper-parameter α. In this part, we first discuss the importance of PCL from the quantitative and visual perspectives and then analyze the role of hyper-parameter α in PCL. MGuid-Net is slightly superior to MGuid-Net$^-$ (i.e., MGuid-Net without PCL) with a 2.7% decline on MAE score and a 0.2% gain on maxF in overall performance, see rows 7 and 8 of Table 2. In addition, Figure 5(c) and (d) display the visualization results of MGuid-Net with and without PCL, respectively. It can be found that although both MGuid-Net and MGuid-Net$^-$ have obtained good detection results, MGuid-Net$^-$ still exists some deficiencies in details relative to MGuid-Net. Briefly, the PCL can help our network to retain more details in predicted saliency maps and further boost the performance of MGuid-Net.



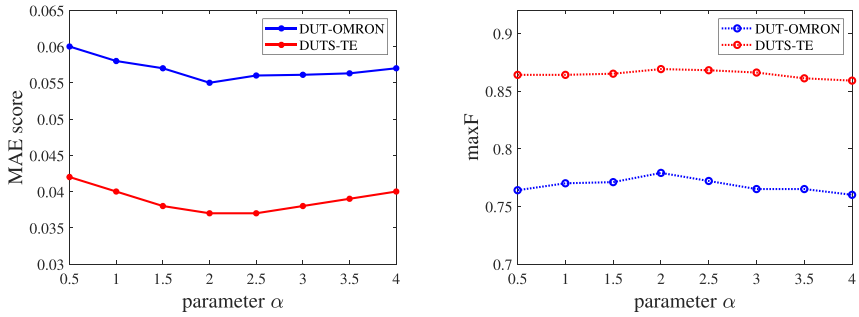
(a) images (b) GT (c) FEU_{before} (d) FEU_{after} (e) GT_{color}

Fig. 4. Visual comparisons of the fused features before and after the FEU enhancement. Here, for ease of observation and comparison, ground truth (GT) and fused features are shown by color image. GT_{color} denotes the color image corresponding to GT, FEU_{before}, and FEU_{after} represent the features before and after the FEU enhancement, respectively.

(a) Images (b) GT (c) MGuid-Net$^-$ (d) MGuid-Net

Fig. 5. Visual comparisons of saliency maps with and without PCL in our model. MGuid-Net$^-$ represents the results without PCL, and MGuid-Net shows the final results with PCL.

To analyze the impact of hyper-parameter α in PCL on model performance, the MGuid-Net are trained on DUTS-TR with different values $\alpha = 0.5, 1, 1.5, \ldots, 4$, and tested on DUT-OMRON and DUTS-TE. The change curves of MAE score and maxF for various α values are depicted in Figure 6. From them, we can find that the MAE scores on both DUT-OMRON and DUTS-TE datasets reach



(a) MAE score (b) maxF

Fig. 6. Effect of parameter α in PCL on model performance.

Table 3. The Quantitative Results (MAE, maxE, maxF, runtime, Params) of MGuid-Net and 10 Compared Models on Five Benchmark Datasets

Dataset	Metric	PAGR	SCRN	BASNet	PiCANet-R	Amulet	EGNet	AFNet	GateNet	DNA	PSGLoss	**Ours**
DUTS-TE	MAE↓	0.055	0.040	0.048	0.040	0.085	0.039	0.046	0.040	0.039	0.038	0.037
	maxE↑	0.895	0.925	0.903	0.915	0.851	0.927	0.910	0.928	0.920	0.927	0.929
	maxF↑	0.817	0.864	0.838	0.840	0.750	0.866	0.839	0.869	0.856	0.868	0.869
DUT-OMRON	MAE↓	0.071	0.056	0.056	0.054	0.098	0.053	0.057	0.055	0.056	0.053	0.055
	maxE↑	0.832	0.875	0.871	0.865	0.834	0.870	0.861	0.876	0.870	0.870	0.871
	maxF↑	0.707	0.772	0.779	0.767	0.715	0.778	0.759	0.781	0.774	0.775	0.779
ECSSD	MAE↓	0.061	0.037	0.037	0.035	0.059	0.037	0.042	0.040	0.035	0.036	0.035
	maxE↑	0.928	0.956	0.951	0.953	0.932	0.955	0.947	0.952	0.952	0.955	0.956
	maxF↑	0.904	0.937	0.931	0.929	0.905	0.936	0.924	0.933	0.934	0.935	0.937
HKU-IS	MAE↓	0.048	0.034	0.032	0.031	0.051	0.031	0.036	0.033	0.032	0.033	0.031
	maxE↑	0.940	0.956	0.951	0.951	0.933	0.958	0.949	0.955	0.957	0.957	0.960
	maxF↑	0.897	0.921	0.919	0.913	0.887	0.924	0.910	0.920	0.925	0.923	0.928
PASCAL-S	MAE↓	0.089	0.063	0.076	0.064	0.100	0.074	0.072	0.067	0.074	0.061	0.061
	maxE↑	0.873	0.910	0.886	0.900	0.862	0.892	0.894	0.904	0.887	0.907	0.910
	maxF↑	0.814	0.856	0.835	0.838	0.805	0.841	0.839	0.848	0.836	0.856	0.857
runtime (s)		—	0.447	2.152	0.752	0.647	2.042	0.726	1.039	0.596	0.566	0.628
Params. (M)		—	25.23	87.06	37.02	33.15	111.64	35.75	128.63	29.31	27.85	27.36

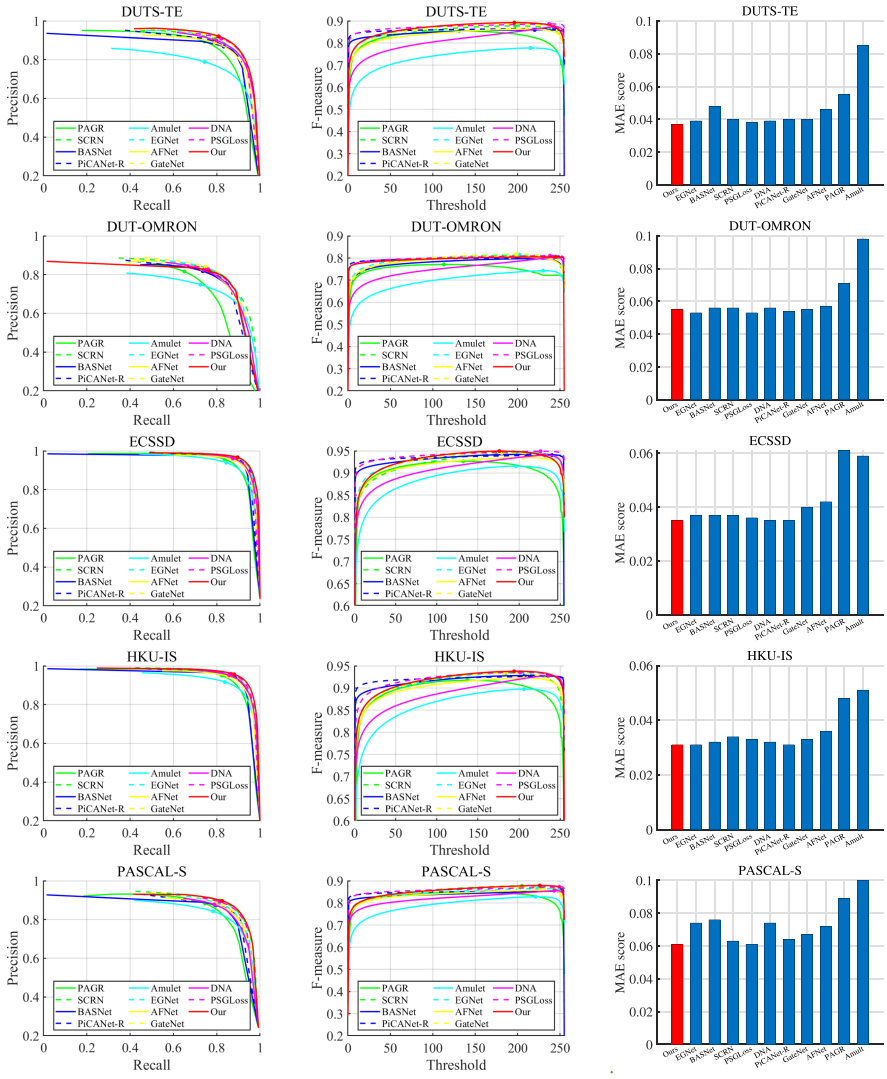
Params (in millions) represents the parameter number of the model. Note that we calculate the runtime (in seconds) of different models 10 times on 352×352 images and report the averaged runtime. Top three scores in each row are shown in red, green, and blue, respectively.

the minimum value near $\alpha = 2$ and then increase monotonically. Meanwhile, the spikes of maxF on DUT-OMRON and DUTS-TE datasets can also be obtained when $\alpha = 2$. Therefore, the value of hyper-parameter α is fixed to 2 in our experiments.

4.4 Comparison with States of the Art

To verify the overall performance of our model, we compare the proposed MGuid-Net with 10 state-of-the-art deep SOD models, including PAGR [54], SCRN [46], BASNet [33], PiCANet-R [27], Amulet [53], EGNet [56], AFNet [11], GateNet [57], DNA [28], and PSGLoss [50]. For fair comparison, the saliency maps of other competing models are directly provided by the authors or generated by running the source codes.

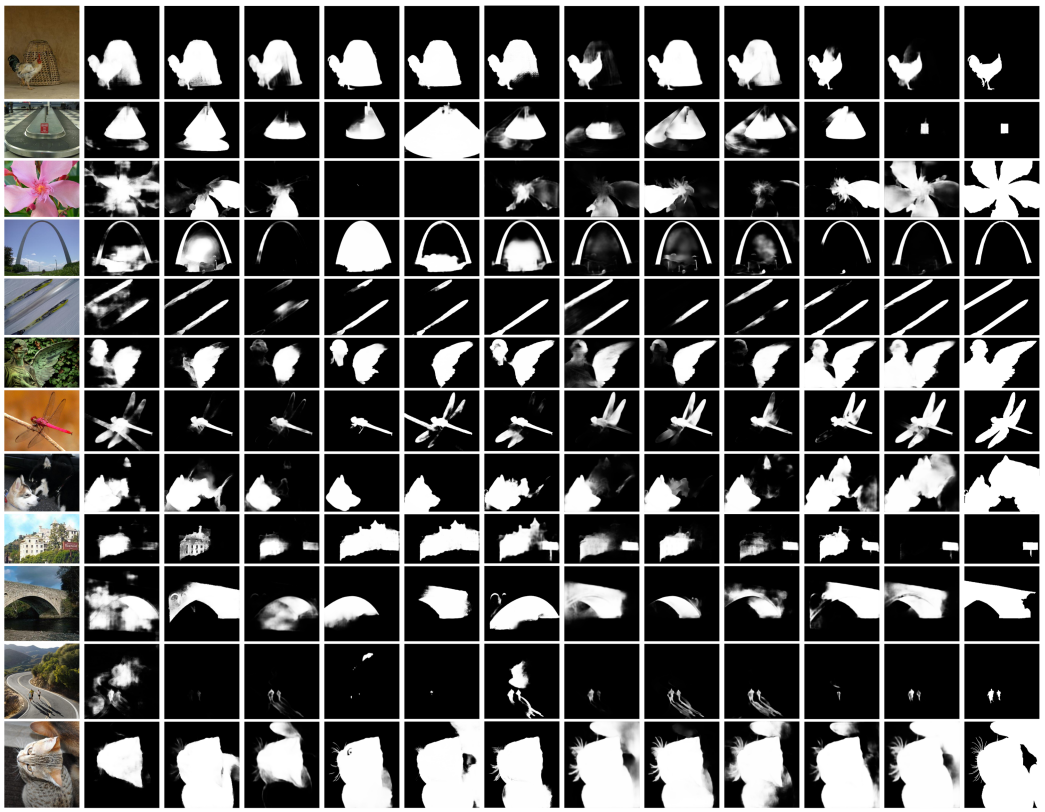
Quantitative comparisons. To compare the above models more comprehensively, both accuracy (MAE, maxE, and maxF) and complexity (runtime, Params) are simultaneously considered in this part. Table 3 lists the quantitative results of different models in the two aspects. For accuracy, it can be found that our model consistently outperforms other compared methods in all metrics



(a) PR Curves (b) maxF Curves (c) MAE Scores

Fig. 7. The quantitative comparisons of 10 state-of-the-art methods and our MGuid-Net on five benchmark datasets. Panels (a)–(c) display the PR curves, maxF curves, and MAE scores, respectively.

on the DUTS-TE, ECSSD, HKU-IS, and PASCAL-S datasets. More specifically, our MGuid-Net outperforms the second best method (i.e., PSGLoss) by 2.9% on MAE score, 0.2% on maxE, and 0.2% on maxF on average quantitative results of the four datasets. On the DUT-OMRON dataset, the results of our model are very close to the best ones achieved by GateNet and are superior or comparable over the results of other methods. In terms of complexity, SCRN, PSGLoss, DNA, and our MGuid-Net are significantly superior to other models. Concretely, the parameter number of our model is the second lowest (27.36M), which is only slightly higher than that of the lowest SCRN (25.23M). Although the proposed MGuid-Net is inferior to SCRN, DNA, and PSGLoss in runtime, its detection performance significantly outperforms others. Note that the runtime of one model is



(a) Images (b) Amulet (c) AFNet (d) PAGR (e) BASNet (f) PiCANet (g) DNA (h) SCRN (i) EGNet (j) GateNet (k) PSGLoss (l) Ours (m) GT

Fig. 8. Visual comparisons of different models in some challenging scenes: cluttered background, small objects, large objects, easily misjudged objects and low contrast between foreground and background.

not only affected by the parameters but also related to the operations used. For example, the multiplication and addition operations with the same parameters consume different times. Moreover, the PR curves, maxF curves, and MAE scores of various models on five public datasets are drawn in Figure 7 to display the quantitative comparisons of these models more intuitively. We can see that the PR curves and maxF curves of MGuid-Net are higher than other curves on all datasets except DUT-OMRON, which also demonstrates that our MGuid-Net gains a competitive performance.

Visual comparisons. Figure 8 depicts the visual comparisons of the aforementioned models in some challenging scenes, including images with a cluttered background (rows 4, 6, and 9), small objects (rows 2, 9, and 11), large objects (rows 3, 6, 8, and 10), easily misjudged objects (rows 1, 2, 5, and 9), and low contrast between the foreground and background (rows 6, 7, 8, and 12). From these results, we can observe that MGuid-Net can detect salient object in different scenes more accurately compared with other algorithms. For the images with easily misjudged objects, our model can correctly select the desired semantics and successfully distribute high saliency values to the salient regions. A typical example is the image in row 9 that contains many houses and a small billboard. Other competing models incorrectly highlight the houses in the background regions, while our model accurately detects the small billboard. For the images with low color contrast, our MGuid-Net can better separate foreground objects from background. As seen the images in row 7, other compared methods either miss the wings of dragonfly or detect the branch

(a) images (b) GT (c) Ours

Fig. 9. Several failure cases of our model.

in background regions as a foreground object. Our MGuid-Net can not only detect more complete dragonfly, but also effectively suppress the background. In short, our model greatly improves the performance of SOD and enhances its robustness by emphasizing semantic information and edge information.

4.5 Failure Cases

Figure 9 displays several failure cases of our model. Similarly to other CNN-based SOD methods, it is still challenging for the proposed MGuid-Net to process the scenario with occluded salient objects. Taking the images in the first row as an example, although the predicted saliency map contains the correct salient objects, the box in the background regions is also misjudged as a foreground object. The main reason is that the semantic information of salient objects is destroyed by the occlusion. A possible remedy is to introduce scene understandings and semantic scores to enhance the ability of our network over semantic selection, which will be investigated in future work.

5 CONCLUSION

In this article, we propose a multi-guidance CNN model for salient object detection called MGuid-Net, which takes full advantage of different level features by various guidance mechanisms to improve SOD performance. Specifically, the self-guidance module and cross-guidance module are used in our model to better refine the extracted multi-level features. An accumulative guidance module with a fusion-enhanced unit adopts a hierarchical structure to integrate refined features more fully. Additionally, we devise a pixelwise contrast loss function that works as an implicit guide to assist our MGuid-Net in capturing more details. Extensive experiments on several popular benchmark datasets demonstrate that our model achieves very competitive performance compared with some state-of-the-art models, and the ablation analyses on DUTS-TE dataset also report the effectiveness of each component in our model.

REFERENCES

[1] Radhakrishna Achanta, Sheila Hemami, Francisco Estrada, and Sabine Susstrunk. 2009. Frequency-tuned salient region detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1597–1604.

[2] Shuhan Chen, Xiuli Tan, Ben Wang, Huchuan Lu, Xuelong Hu, and Yun Fu. 2020. Reverse attention-based residual network for salient object detection. *IEEE Trans. Image Process.* 29 (2020), 3763–3776.

[3] Ming-Ming Cheng and Deng-Ping Fan. 2021. Structure-measure: A new way to evaluate foreground maps. *Int. J. Comput. Vis.* 129, 9 (2021), 2622–2638.
[4] Ming-Ming Cheng, Niloy J. Mitra, Xiaolei Huang, Philip H. S. Torr, and Shi-Min Hu. 2015. Global contrast based salient region detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 37, 3 (2015), 569–582.
[5] Ming-Ming Cheng, Jonathan Warrell, Wen-Yan Lin, Shuai Zheng, Vibhav Vineet, and Nigel Crook. 2013. Efficient salient region detection with soft image abstraction. In *Proceedings of the IEEE International Conference on Computer Vision*. 1529–1536.
[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 248–255.
[7] Bo Dong, Yan Zhou, Chuanfei Hu, Keren Fu, and Geng Chen. 2021. BCNet: Bidirectional collaboration network for edge-guided salient object detection. *Neurocomputing* 437 (2021), 58–71.
[8] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. 2010. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.* 88, 2 (2010), 303–338.
[9] Deng-Ping Fan, Ming-Ming Cheng, Yun Liu, Tao Li, and Ali Borji. 2017. Structure-measure: A new way to evaluate foreground maps. In *Proceedings of the IEEE International Conference on Computer Vision*. 4548–4557.
[10] Deng-Ping Fan, Cheng Gong, Yang Cao, Bo Ren, Ming-Ming Cheng, and Ali Borji. 2018. Enhanced-alignment measure for binary foreground map evaluation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 698–704.
[11] Mengyang Feng, Huchuan Lu, and Errui Ding. 2019. Attentive feedback network for boundary-aware salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1623–1632.
[12] Wenlong Guan, Tiantian Wang, Jinqing Qi, Lihe Zhang, and Huchuan Lu. 2019. Edge-aware convolution neural network based salient object detection. *IEEE Sign. Process. Lett.* 26, 1 (2019), 114–118.
[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.
[14] Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, and Philip H. S. Torr. 2019. Deeply supervised salient object detection with short connections. *IEEE Trans. Pattern Anal. Mach. Intell.* 41, 4 (2019), 815–828.
[15] Xiaowei Hu, Lei Zhu, Jing Qin, Chi-Wing Fu, and Pheng-Ann Heng. 2018. Recurrently aggregating deep features for salient object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 6943–6950.
[16] Dominik A. Klein and Simone Frintrop. 2011. Center-surround divergence of feature statistics for salient object detection. In *Proceedings of the IEEE International Conference on Computer Vision*. 2214–2219.
[17] Jason Kuen, Zhenhua Wang, and Gang Wang. 2016. Recurrent attentional networks for saliency detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3668–3677.
[18] Gayoung Lee, Yu-Wing Tai, and Junmo Kim. 2016. Deep saliency with encoded low level distance map and high level features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 660–668.
[19] Guanbin Li and Yizhou Yu. 2015. Visual saliency based on multiscale deep features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5455–5463.
[20] Kunpeng Li, Ziyan Wu, Kuan-Chuan Peng, Jan Ernst, and Yun Fu. 2020. Guided attention inference network. *IEEE Trans. Pattern Anal. Mach. Intell.* 42, 12 (2020), 2996–3010.
[21] Yin Li, Xiaodi Hou, Christof Koch, James M. Rehg, and Alan L. Yuille. 2014. The secrets of salient object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 280–287.
[22] Guibiao Liao, Wei Gao, Qiuping Jiang, Ronggang Wang, and Ge Li. 2020. MMNet: Multi-stage and multi-scale fusion network for RGB-D salient object detection. In *Proceedings of the 28th ACM International Conference on Multimedia*. 24362444.
[23] Feng Lin, Wengang Zhou, Jiajun Deng, Bin Li, Yan Lu, and Houqiang Li. 2021. Residual refinement network with attribute guidance for precise saliency detection. *ACM Trans. Multimedia Comput. Commun. Appl.* 17, 3, Article 81 (2021), 19 pages.
[24] Jiang-Jiang Liu, Qibin Hou, Ming-Ming Cheng, Jiashi Feng, and Jianmin Jiang. 2019. A simple pooling-based design for real-time salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3912–3921.
[25] Jiang-Jiang Liu, Zhi-Ang Liu, Pai Peng, and Ming-Ming Cheng. 2021. Rethinking the U-shape structure for salient object detection. *IEEE Trans. Image Process.* 30 (2021), 9030–9042.
[26] Nian Liu and Junwei Han. 2016. DHSNet: Deep hierarchical saliency network for salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 678–686.
[27] Nian Liu, Junwei Han, and Ming-Hsuan Yang. 2018. PiCANet: Learning pixel-wise contextual attention for saliency detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3089–3098.
[28] Yun Liu, Ming-Ming Cheng, Xin-Yu Zhang, Guang-Yu Nie, and Meng Wang. 2022. DNA: Deeply supervised nonlinear aggregation for salient object detection. *IEEE Trans. Cybernet.* 52, 7 (2022), 6131–6142.

[29] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3431–3440.

[30] Shanmei Lu, Qiang Guo, and Yongxia Zhang. 2020. Salient object detection using recurrent guidance network with hierarchical attention features. *IEEE Access* 8 (2020), 151325–151334.

[31] Youwei Pang, Xiaoqi Zhao, Lihe Zhang, and Huchuan Lu. 2020. Multi-scale interactive network for salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 9410–9419.

[32] Federico Perazzi, Philipp Krähenbühl, Yael Pritch, and Alexander Hornung. 2012. Saliency filters: Contrast based filtering for salient region detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 733–740.

[33] Xuebin Qin, Zichen Zhang, Chenyang Huang, Chao Gao, Masood Dehghan, and Martin Jagersand. 2019. BASNet: Boundary-aware salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7471–7481.

[34] Ran Shi, Jing Ma, King Ngi Ngan, Jian Xiong, and Tong Qiao. 2022. Objective object segmentation visual quality evaluation: Quality measure and pooling method. *ACM Trans. Multimedia Comput. Commun. Appl.* 18, 3, Article 73 (2022), 19 pages.

[35] Wei-Chih Tu, Shengfeng He, Qingxiong Yang, and Shao-Yi Chien. 2016. Real-time salient object detection with a minimum spanning tree. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2334–2342.

[36] Jingdong Wang, Huaizu Jiang, Zejian Yuan, Ming-Ming Cheng, Xiaowei Hu, and Nanning Zheng. 2017. Salient object detection: A discriminative regional feature integration approach. *Int. J. Comput. Vis.* 123, 2 (2017), 251–268.

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

[38] Linzhao Wang, Lijun Wang, Huchuan Lu, Pingping Zhang, and Xiang Ruan. 2018. Salient object detection with recurrent fully convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 41, 7 (2018), 1734–1746.

[39] Tiantian Wang, Lihe Zhang, Shuo Wang, Huchuan Lu, Gang Yang, Xiang Ruan, and Ali Borji. 2018. Detect globally, refine locally: A novel approach to saliency detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3127–3135.

[40] Wenguan Wang, Qiuxia Lai, Huazhu Fu, Jianbing Shen, Haibin Ling, and Ruigang Yang. 2022. Salient object detection in the deep learning era: An in-depth survey. *IEEE Trans. Pattern Anal. Machine Intell.* 44, 6 (2022), 3239–3259.

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

[42] Wenguan Wang, Jianbing Shen, Xingping Dong, Ali Borji, and Ruigang Yang. 2019. Inferring salient objects from human fixations. *IEEE Trans. Pattern Anal. Mach. Intell.* 42, 8 (2019), 1913–1927.

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

[44] Yupei Wang, Xin Zhao, Xuecai Hu, Yin Li, and Kaiqi Huang. 2019. Focal boundary guided salient object detection. *IEEE Trans. Image Process.* 28, 6 (2019), 2813–2824.

[45] Hongfa Wen, Chenggang Yan, Xiaofei Zhou, Runmin Cong, Yaoqi Sun, Bolun Zheng, Jiyong Zhang, Yongjun Bao, and Guiguang Ding. 2021. Dynamic selective network for RGB-D salient object detection. *IEEE Trans. Image Process.* 30 (2021), 9179–9192.

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

[47] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. 2010. SUN database: Large-scale scene recognition from abbey to zoo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3485–3492.

[48] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. 2013. Hierarchical saliency detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1155–1162.

[49] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. 2013. Saliency detection via graph-based manifold ranking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3166–3173.

[50] Sheng Yang, Weisi Lin, Guosheng Lin, Qiuping Jiang, and Zichuan Liu. 2021. Progressive self-guided loss for salient object detection. *IEEE Trans. Image Process.* 30 (2021), 8426–8438.

[51] Jun Zhang, Meng Wang, Liang Lin, Xun Yang, Jun Gao, and Yong Rui. 2017. Saliency detection on light field: A multi-cue approach. *ACM Trans. Multimedia Comput. Commun. Appl.* 13, 3, Article 32 (2017), 22 pages.

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

[53] Pingping Zhang, Dong Wang, Huchuan Lu, Hongyu Wang, and Xiang Ruan. 2017. Amulet: Aggregating multi-level convolutional features for salient object detection. In *Proceedings of the IEEE International Conference on Computer Vision*. 202–211.

[54] Xiaoning Zhang, Tiantian Wang, Jinqing Qi, Huchuan Lu, and Gang Wang. 2018. Progressive attention guided recurrent network for salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 714–722.

[55] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3927–3936.

[56] Jia-Xing Zhao, Jiang-Jiang Liu, Deng-Ping Fan, Yang Cao, Jufeng Yang, and Ming-Ming Cheng. 2019. EGNet: Edge guidance network for salient object detection. In *Proceedings of the IEEE International Conference on Computer Vision*. 8778–8787.

[57] Xiaoqi Zhao, Youwei Pang, Lihe Zhang, Huchuan Lu, and Lei Zhang. 2020. Suppress and balance: A simple gated network for salient object detection. In *Proceedings of the European Conference on Computer Vision*. 35–51.

[58] Zhirui Zhao, Changqun Xia, Chenxi Xie, and Jia Li. 2021. Complementary trilateral decoder for fast and accurate salient object detection. In *Proceedings of the 29th ACM International Conference on Multimedia*. 4967–4975.

[59] 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 *Proceedings of the IEEE International Conference on Multimedia and Expo*. 199–204.

[60] Yunzhi Zhuge, Gang Yang, Pingping Zhang, and Huchuan Lu. 2018. Boundary-guided feature aggregation network for salient object detection. *IEEE Sign. Process. Lett.* 25, 12 (2018), 1800–1804.

Received 30 April 2022; revised 25 October 2022; accepted 30 October 2022