Multi-scale Interactive Network for Salient Object Detection

Youwei Pang[1†], Xiaoqi Zhao[1†], Lihe Zhang[1*] and Huchuan Lu[1,2]
[1]Dalian University of Technology, China
[2]Peng Cheng Laboratory
{lartpang, zxq}@mail.dlut.edu.cn, {zhanglihe, lhchuan}@dlut.edu.cn

Abstract

Deep-learning based salient object detection methods achieve great progress. However, the variable scale and unknown category of salient objects are great challenges all the time. These are closely related to the utilization of multi-level and multi-scale features. In this paper, we propose the aggregate interaction modules to integrate the features from adjacent levels, in which less noise is introduced because of only using small up-/down-sampling rates. To obtain more efficient multi-scale features from the integrated features, the self-interaction modules are embedded in each decoder unit. Besides, the class imbalance issue caused by the scale variation weakens the effect of the binary cross entropy loss and results in the spatial inconsistency of the predictions. Therefore, we exploit the consistency-enhanced loss to highlight the fore-/back-ground difference and preserve the intra-class consistency. Experimental results on five benchmark datasets demonstrate that the proposed method without any post-processing performs favorably against 23 state-of-the-art approaches. The source code will be publicly available at https://github.com/lartpang/MINet.

1. Introduction

Salient object detection (SOD) aims at distinguishing the most visually obvious regions. It is growing rapidly with the help of data-driven deep learning methods and has been applied in many computer vision fields, such as visual tracking [24], image retrieval [10], non-photorealistic rendering [28], 4D saliency detection [33], no-reference synthetic image quality assessmen [38] and so on. Although great progress has been made at present, two issues still need to be paid attention to *how to extract more effective information from the data of scale variation* and *how to improve the spatial coherence of predictions in this situation*. Due to various scales of salient regions, the CNN-based methods, which are limited by the absence of necessary detailed information owing to the repeated sub-sampling, have difficulty in consistently and accurately segmenting salient objects of different scales (Fig. 1). In addition, on account of the inherent localization of convolution operation and the pixel-level characteristics of the cross entropy function, it is difficult to achieve uniform highlighting of objects.

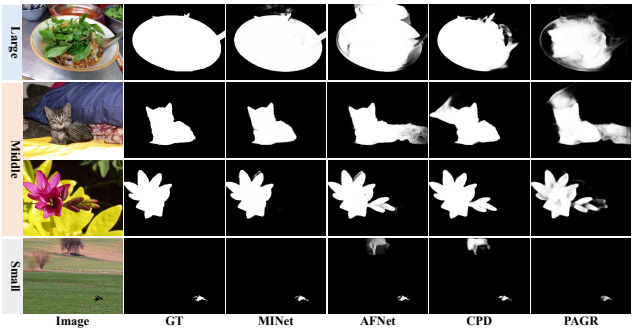

Figure 1. Several visual examples with size-varying objects and their predictions generated by the proposed MINet, AFNet [9], CPD [41] and PAGR [53] methods.

For the first problem, the main solution of the existing methods is to layer-by-layer integrate shallower features. Some methods [23, 53, 4, 9, 40, 41, 27, 37] connect the features at the corresponding level in the encoder to the decoder via the transport layer (Fig. 2(a, c, e)). The single-level features can only characterize the scale-specific information. In the top-down pathway, the representation capability of details in shallow features is weakened due to the continuous accumulation of the deeper features. To utilize the multi-level features, some approaches [51, 13, 34] combine the features from multiple layers in a fully-connected manner or a heuristic style (Fig. 2(b, f, g)). However, integrating excessive features and lacking a balance between different resolutions easily lead to high computational cost, plenty of noise and fusion difficulties, thereby disturbing the subsequent information recovery in the top-down pathway. Moreover, the atrous spatial pyramid pooling module

[†]These authors contributed equally to this work.
[*]Corresponding author.

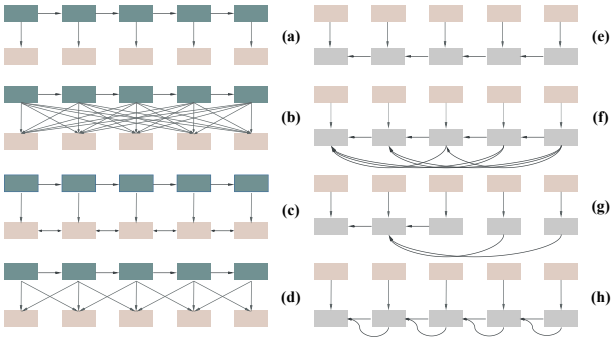

Figure 2. Illustration of different architectures. Green blocks, orange blocks and gray blocks respectively denote the different convolutional blocks in the encoder, the transport layer and the decoder. Left column: the connection patterns between the encoder and the transport layer; Right column: the connection patterns between the transport layer and the decoder. (a, e) FCN [22]; (b) Amulet [51]; (c) BMPM [48]; (d) AIMs (Sec. 3.2); (f) DSS [13]; (g) DGRL [34]; (h) SIMs (Sec. 3.3).

(ASPP) [3] and the pyramid pooling module (PPM) [55] are used to extract multi-scale context-aware features and enhance the single-layer representation [6, 32]. Nonetheless, the existing methods usually equip these modules behind the encoder, which results in that their networks miss many necessary details due to the limitation of the low resolution of the top-layer features. For the second problem, some existing models [41, 27] mainly employ a specific branch or an additional network to refine the results. Nevertheless, these methods are faced with the problem of computational redundancy and training difficulties, which is not conducive to further applications.

Inspired by the idea of the mutual learning proposed by Zhang et al. [54], we propose an aggregate interaction strategy (AIM) to make better use of multi-level features and avoid the interference in feature fusion caused by large resolution differences (Fig. 2(d)). We collaboratively learn knowledge guidance to effectively integrate the contextual information from adjacent resolutions. To further obtain abundant scale-specific information from the extracted features, we design a self-interaction module (SIM) (Fig. 2(h)). Two interactive branches of different resolutions are trained to learn multi-scale features from a single convolutional block. AIMs and SIMs effectively improve the ability to deal with scale variations in the SOD task. Unlike the settings in [54], in the two modules, the mutual learning mechanism is incorporated into feature learning. Each branch can more flexibly integrate information from other resolutions through interactive learning. In AIMs and SIMs, the main branch (B^1 in Fig. 4 and B^0 in Fig. 5) is supplemented by the auxiliary branches and its discriminating power is further enhanced. In addition, the multi-scale issue also causes a serious imbalance between foreground and background regions in the datasets, hence we embed a consistency-enhanced loss (CEL) into the training stage, which is not sensitive to the scale of objects. At the same time, the CEL can better handle the spatial coherence issue and uniformly highlight salient regions without additional parameters, because its gradient has the characteristics of keeping intra-class consistency and enlarging inter-class differences.

Our contributions are summarized as three folds:

- We propose the MINet to effectively meet scale challenges in the SOD task. The aggregate interaction module can efficiently utilize the features from adjacent layers by the way of mutual learning and the self-interaction module makes the network adaptively extract multi-scale information from data and better deal with scale variation.
- We utilize the consistency-enhanced loss as an assistant to push our model to uniformly highlight the entire salient region and better handle the pixel imbalance problem between fore- and back-ground regions caused by various scales of objects, without any post-processing or extra parameters.
- We compare the proposed method with 23 state-of-the-art SOD methods on five datasets. It achieves the best performance under different evaluation metrics. Besides, the proposed model has a forward reasoning speed of 86.1 FPS on GPU.

2. Related Work

2.1. Salient Object Detection

Early methods are mainly based on hand-crafted priors [5, 39, 49, 47]. Their generalization and effectiveness are limited. The early deep salient object detection (SOD) methods [57, 16] use the multi-layer perception to predict the saliency score for each processing unit of an image. These methods have low computational efficiency and damage the potential feature structure. See [2, 35] for more details about traditional and early deep methods.

Recently, some methods [20, 53] introduce the fully convolutional network (FCN) [22] and achieve promising results. Moreover, Liu et al. [20] hierarchically embed global and local context modules into the top-down pathway which constructs informative contextual features for each pixel. Chen et al. [4] propose reverse attention in the top-down pathway to guide residual saliency learning, which drives the network to discover complement object regions and details. Nonetheless, the above-mentioned methods only employ individual resolution features in each decoder unit, which is not an effective enough strategy to cope with complex and various scale problems.

2.2. Scale Variation

Scale variation is one of the major challenges in the SOD task. Limited by the localized convolution operation and

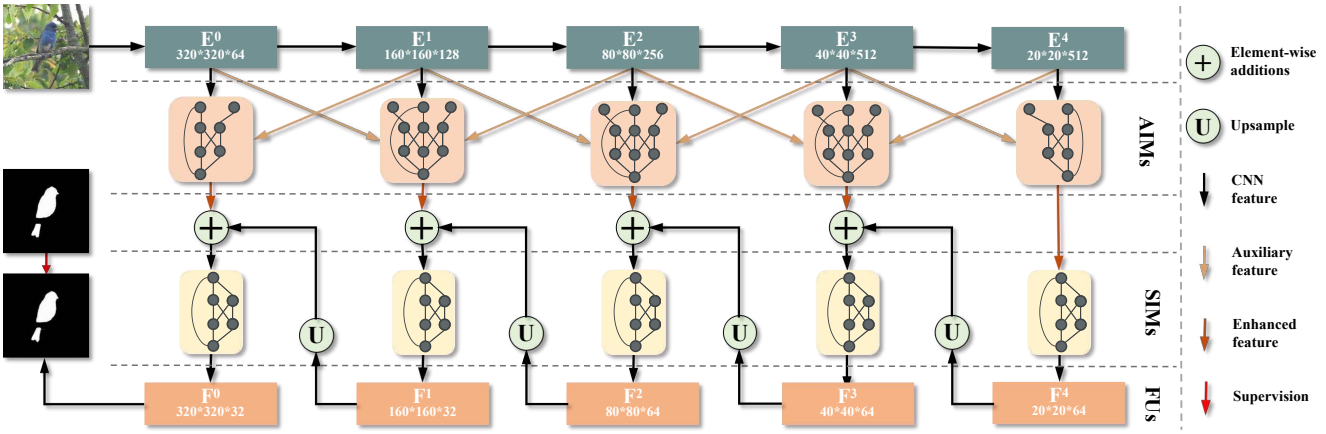

Figure 3. The overall framework of the proposed model. Each colorful box represents a feature processing module. Our model takes a RGB image ($320 \times 320 \times 3$) as input, and exploits VGG-16 [29] blocks $\{\mathbf{E}^i\}_{i=0}^4$ to extract multi-level features. The features are integrated by AIMs ($\{\mathbf{AIM}^i\}_{i=0}^4$) and then, the outputted features are gradually combined by using SIMs ($\{\mathbf{SIM}^i\}_{i=0}^4$) and fusion units ($\{\mathbf{F}^i\}_{i=0}^4$) to generate the final prediction $\mathcal{P}$ supervised by the ground truth $\mathcal{G}$.

sub-sampling operation, it is difficult for CNN to handle this problem. On one hand, the amount of information about objects, which are embedded in the features of different resolutions, changes with the scale of objects. A straightforward strategy is to roughly integrate all features. On the other hand, each convolutional layer only has the capability of processing a special scale. Therefore, we need to characterize the multi-scale information from a single layer by building a multi-path feature extraction structure.

Multi-level Information. Zhang et al. [51] simply combine all level features into the transport layer. This kind of coarse fusion easily produces information redundancy and noise interference. In [48], a gate function is exploited to control the message passing rate to optimize the quality of information exchange between layers. Nevertheless, multiple gating processing leads to severe attenuation of the information from other layers, which limits the learning ability of the network. Different from these methods, we only fuse the features from the adjacent layers by reason of their closer degree of abstraction and concurrently obtain abundant scale information.

Multi-scale Information. The atrous spatial pyramid pooling (ASPP) [3] and the pyramid pooling module (PPM) [55] are two common choices for multi-scale information extraction and are often fixed at the deepest level in the network [6, 32]. Since the deeper features contain less information about small-scale objects, which is especially true for the top-layer features, these methods can not effectively deal with large scale variation. Besides, in [37], the pyramid attention module can obtain multi-scale attention maps to enhance features through multiple downsampling and softmax operations on all positions. But such a softmax severely suppresses non-maximum values and is more sensitive to noise. It does not improve the scale issue well. To avoid misjudging small objects, we propose a multi-scale processing module where two branches interactively learn features. Through data-driven training, the two-path structure can learn rich multi-scale representation. In addition, the oversized and undersized objects cause the imbalance between foreground and background samples, which weakens the effect of pixel-level supervision. We introduce the consistency-enhanced loss (CEL) as an aid to the cross entropy loss. The CEL is not sensitive to the size of objects. It can overcome the difficulties of supervision and perform very well in the face of large scale variation.

2.3. Spatial Coherence

To improve spatial coherence and quality of saliency maps, some non-deep methods often integrate an oversegmentation process that generates regions [44], superpixels [45], or object proposals [11]. For deep learning based methods, Wu et al. [41] propose a cascaded partial decoder framework with two branches and directly utilize attention maps generated by the attention branch to refine the features from the saliency detection branch. Qin et al. [27] employ a residual refinement module combined with a hyper loss to further refine the predictions, which significantly reduces the inference speed. In this paper, the CEL pays more attention to the overall effect of the prediction. It helps obtain a more uniform saliency result and is a better trade-off between the effect and the speed.

3. Proposed Method

In this paper, we propose an interactive integration network which fuses multi-level and multi-scale feature information to deal with the prevalent scale variation issue in saliency object detection (SOD) task. The overall network structure is shown in Fig. 3. Encoder blocks, ag-

gregate interaction modules (AIMs), self-interaction modules (SIMs) and fusion units (FUs) are denoted as $\{\mathbf{E}^i\}_{i=0}^4$, $\{\mathbf{AIM}^i\}_{i=0}^4$, $\{\mathbf{SIM}^i\}_{i=0}^4$ and $\{\mathbf{F}^i\}_{i=0}^4$, respectively.

3.1. Network Overview

Our model is built on the FCN architecture with the pre-trained VGG-16 [29] or ResNet-50 [12] as the backbone, both of which only retain the feature extraction network. Specifically, we remove the last max-pooling layer of the VGG-16 to maintain the details of the final convolutional layer. Thus, the input is sub-sampled with a factor of 16 for the VGG-16 and with a factor of 32 for the ResNet-50. We use the backbone to extract multi-level features and abstractions, and then each AIM (Fig. 4) uses the features of adjacent layers as the input to efficiently employ the multi-level information and provide more relevant and effective supplementary for the current resolution. Next, in the decoder, every SIM (Fig. 5) is followed by an FU which is a combination of a convolutional layer, a batch normalization layer and a ReLU layer. The SIM can adaptively extract multi-scale information from the specific levels. The information is further integrated by the FU and fed to the shallower layer. In addition, we introduce the consistency-enhanced loss as an auxiliary loss to supervise the training stage. In this section, we will introduce these modules in detail. To simplify the description, all subsequent model parameters are based on the VGG-16 backbone.

3.2. Aggregate Interaction Module

In the feature extraction network, different levels of convolutional layers correspond to a different degree of feature abstraction. The multi-level integration can enhance the representation ability of different resolution features: 1) In the shallow layers, the detailed information can be further strengthened and the noise can be suppressed; 2) In the middle layers, both semantic and detailed information is taken into account at the same time, and the proportion of different abstraction information in the features can be adaptively adjusted according to the needs of the network itself, thereby achieving more flexible feature utilization; 3) In the top layer, richer semantic information can be mined when considering adjacent resolutions. In particular, we propose the aggregate interaction module (AIM) (Fig. 4) to aggregate features by a strategy of interactive learning.

The i^{th} AIM is denoted as $\mathbf{AIM}^i$, the input of which consists of features f_e^{i-1}, f_e^i and f_e^{i+1} from the encoder, as shown in Fig. 4 (b). After the initial **transformation** by a combination of a single convolutional layer, a batch normalization layer and a ReLU layer, the channel number of these features is reduced. In the **interaction** stage, the B^0 branch and the B^2 branch are adjusted by the pooling, neighbor interpolation and convolution operations, and then both of them are merged into the B^1 branch by the element-wise addition. At the same time, the B^1 branch is also adjusted its resolution and is respectively combined into the B^0 and B^2 branches. Finally, the three branches are **fused** together through the subsequent convolutional layer and the channel number is also reduced. In order to efficiently train the AIMs and increase the weight of f_e^i to ensure that other branches only act as supplements, a residual learning strategy is introduced. The outputted feature is denoted as $f_{AIM}^i \in \mathbb{R}^{N_i \times H_i \times W_i \times C_i}$, where $C_0 = 32$ and $C_{i\neq 0} = 64$. For $\mathbf{AIM}^0$ and $\mathbf{AIM}^4$, their inputs only contain f_e^0, f_e^1 and f_e^3, f_e^4, correspondingly (Fig. 4 (a, c)). The entire process is formulated as follows:

$$
\begin{aligned}
f_{AIM}^i &= \mathbf{I}_{AIM}^i(f_e^i) + \mathbf{M}_{AIM}^i(f_{AB}^i), \\
f_{AB}^i &= \begin{cases} \sum_{j=1}^{2} \mathbf{B}_{AIM}^{i,j}(f_e^{j-1}) & \text{if } i = 0, \\ \sum_{j=0}^{2} \mathbf{B}_{AIM}^{i,j}(f_e^{i+j-1}) & \text{if } i = 1,2,3, \\ \sum_{j=0}^{1} \mathbf{B}_{AIM}^{i,j}(f_e^{i+j-1}) & \text{if } i = 4, \end{cases}
\end{aligned} \tag{1}
$$

where $\mathbf{I}(\cdot)$ and $\mathbf{M}(\cdot)$ represent the identity mapping and the branch merging, respectively. $\mathbf{B}_{AIM}^{i,j}(\cdot)$ is the overall operation of the j^{th} branch (i.e. B^j) in the $\mathbf{AIM}^i$. Due to space constraints, please refer to Fig. 4 for the computational details inside each branch.

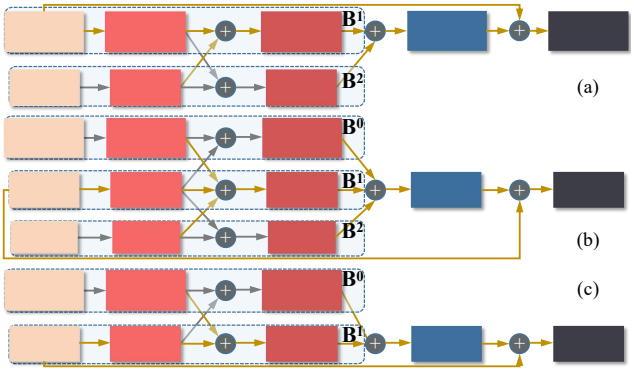

Figure 4. Illustration of aggregate interaction modules (AIMs). $\mathbf{B}^i$: All operations in the i^{th} branch B^i.

3.3. Self-Interaction Module

The AIMs aim at achieving efficient utilization of the inter-layer convolutional features, while the self-interaction modules (SIMs) are proposed to produce multi-scale representation from the intra-layer features. The details of the SIMs can be seen in Fig. 5. Similarly, we also apply the **transformation-interaction-fusion** strategy in the SIMs. Concretely speaking, the resolution and dimension of the input feature are reduced by a convolutional layer, at first. In each branch, the SIM performs an initial **transformation** to adapt to the following interaction operation: We up-sample low-resolution features and sub-sample high-resolution features to the same resolution as the features from the other branch. The **interaction** between high- and

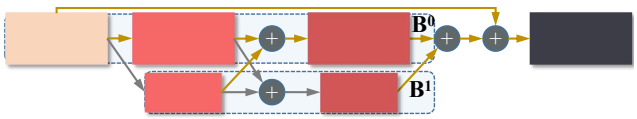

Figure 5. Illustration of self-interaction modules (SIMs). $\mathbf{B}^i$: All operations in the i^{th} branch B^i.

low-resolution features with different channel numbers can obtain plenty of knowledge about various scales and maintain high-resolution information with a low parameter quantity. For ease of optimization, a residual connection is also adopted as shown in Fig. 5. After up-sampling, normalization, and nonlinear processing, an FU is used to **fuse** the features of double paths from the SIM and the residual branch. Integrating the SIMs into the decoder allows the network to adaptively deal with scale variation of different samples during the training stage. The entire process is written as:

$$f_{SIM}^i = f_{add}^i + \mathbf{M}_{SIM}^i \left(\mathbf{B}_{SIM}^{i,0}(f_{add}^i) + \mathbf{B}_{SIM}^{i,1}(f_{add}^i) \right), \tag{2}$$

where f_{SIM}^i is the output of the $\mathbf{SIM}^i$. $\mathbf{M}(\cdot)$ represents the branch merging and $\mathbf{B}_{SIM}^{i,j}$ denotes the operation in the j^{th} branch (i.e. B^j) in the $\mathbf{SIM}^i$, and the input feature f_{add}^i is calculated as follows:

$$f_{add}^i = \begin{cases} f_{AIM}^i + \mathbf{U}^{i+1}(\mathbf{F}^{i+1}(f_{SIM}^{i+1})) & \text{if } i = 0, 1, 2, 3, \\ f_{AIM}^i & \text{if } i = 4, \end{cases} \tag{3}$$

where $\mathbf{U}^{i+1}(\cdot)$ and $\mathbf{F}^{i+1}(\cdot)$ denote the $(i+1)^{th}$ up-sampling operation and fusion unit in the top-down pathway. For more details about the SIMs, please see Fig. 5.

3.4. Consistency-Enhanced Loss

In the SOD task, the widely used binary cross entropy function accumulates the per-pixel loss in the whole batch and does not consider the inter-pixel relationships, which can not explicitly prompt the model to highlight the foreground region as smoothly as possible and deal well with the sample imbalance issue. To this end, we propose a consistency-enhanced loss (CEL). First of all, the final prediction is calculated as follows:

$$\mathcal{P} = \text{Sigmoid}(\text{Conv}(\mathbf{F}^0(f_{add}^0))), \tag{4}$$

where $\mathcal{P} \in \mathbb{R}^{N \times H \times W \times 1}$ denotes N saliency maps in a batch, and N is the batchsize. $0 < p \in \mathcal{P} < 1$ is the probability of belonging to salient regions. Sigmoid(Conv$(\cdot)$) actually represents the last convolutional layer with a nonlinear activation function in the decoder. The binary cross entropy loss (BCEL) function is written as follows:

$$L_{BCEL} = \sum_{p \in \mathcal{P}, g \in \mathcal{G}} -\left[g \log p + (1-g) \log(1-p) \right], \tag{5}$$

where $\log(\cdot)$ is also an element-wise operation. $\mathcal{G} \in \{0,1\}^{N \times H \times W \times 1}$ represents the ground truth. To address the fore-/back-ground imbalance issue caused by various scales, the loss function needs to meet two requirements, at least: 1) It focuses more on the foreground than the background, and the difference at the scale of objects does not induce the wide fluctuation in the computed loss; 2) When the predicted foreground region is completely disjoint from the ground-truth one, there should be the largest penalty. Based on the two points, we consider the topological relationships among regions to define the CEL as follows:

$$\begin{aligned} L_{CEL} &= \frac{|FP + FN|}{|FP + 2TP + FN|} \\ &= \frac{\sum(p - pg) + \sup(g - pg)}{\sum p + \sum g}, \end{aligned} \tag{6}$$

where TP, FP and FN represent true-positive, false-positive and false-negative, respectively. $|\cdot|$ computes the area. $FP + FN$ denotes the difference set between the union and intersection of the predicted foreground region and the ground-truth one, while $FP + 2TP + FN$ represents the sum of this union set and this intersection. When $\{p|p > 0, p \in \mathcal{P}\} \cap \{g|g = 1, g \in \mathcal{G}\} = \varnothing$, the loss reaches its maximum, i.e. $L_{CEL} = 1$. Since p is continuous, L_{CEL} is differentiable with reference to p. Thus, the network can be trained in an end-to-end manner.

To compare L_{CEL} with L_{BCEL}, we analyze their gradients which directly act on the network predictions. Their derivatives are expressed as follows:

$$\frac{\partial L_{BCEL}}{\partial p} = -\frac{g}{p} + \frac{1-g}{1-p}, \tag{7}$$

$$\frac{\partial L_{CEL}}{\partial p} = \frac{1-2g}{\sum(p+g)} - \frac{\sum(p+g-2pg)}{[\sum(p+g)]^2}. \tag{8}$$

It can be observed that $\partial L_{BCEL}/\partial p$ only relies on the prediction of the individual position. While $\partial L_{CEL}/\partial p$ is related to all pixels in both the prediction $\mathcal{P}$ and the ground truth $\mathcal{G}$. Therefore, the CEL is considered to enforce a global constraint on the prediction results, which can produce more effective gradient propagation. In Equ. (8), except that the numerator term $1 - 2g$ is position-specific, the other terms are image-specific. And this numerator is closely related to the binary ground truth, which results in that the inter-class derivatives have large differences while the intra-class ones are relatively consistent. This has several merits: 1) It ensures that there is enough large gradient to drive the network in the later stage of training; 2) It helps solve the intra-class inconsistency and inter-class indistinction issues, to some extent, thereby promoting the predicted boundaries of salient objects to become sharper. Finally, the total loss function can be written as:

$$L = L_{BCEL}(\mathcal{P}, \mathcal{G}) + \lambda L_{CEL}(\mathcal{P}, \mathcal{G}), \tag{9}$$

Table 1. Quantitative evaluation. The maximum and mean F-measure (larger is better), S-measure (larger is better) and MAE (smaller is better) of different saliency detection methods on five benchmark datasets. The best three results are highlighted in **red**, **green** and **blue**. $*$: with post-processing; $X101$: ResNeXt-101 [43] as backbone; $R101$: ResNet-101 [12] as backbone.

Model	DUTS-TE						DUT-OMRON						HKU-IS						ECSSD						Pascal-S					
	F_{max}	F_{avg}	F^{ω}_{β}	E_m	S_m	MAE	F_{max}	F_{avg}	F^{ω}_{β}	E_m	S_m	MAE	F_{max}	F_{avg}	F^{ω}_{β}	E_m	S_m	MAE	F_{max}	F_{avg}	F^{ω}_{β}	E_m	S_m	MAE	F_{max}	F_{avg}	F^{ω}_{β}	E_m	S_m	MAE
VGG-16																														
Ours	**0.877**	**0.823**	**0.813**	**0.912**	**0.875**	**0.039**	**0.794**	**0.741**	**0.719**	**0.864**	0.822	**0.057**	**0.932**	**0.906**	**0.892**	**0.955**	**0.914**	**0.030**	**0.943**	**0.922**	**0.905**	**0.947**	**0.919**	**0.036**	**0.882**	**0.843**	**0.820**	**0.898**	**0.855**	**0.065**
EGNet$_{19}$	**0.877**	**0.800**	**0.797**	**0.895**	**0.878**	**0.044**	**0.809**	**0.744**	**0.728**	**0.864**	**0.836**	**0.057**	**0.927**	0.893	**0.875**	**0.950**	**0.910**	**0.035**	**0.943**	**0.913**	**0.892**	0.941	**0.919**	**0.041**	**0.871**	0.821	0.798	0.873	0.848	0.078
AFNet$_{19}$	**0.863**	0.793	0.785	**0.895**	**0.867**	0.046	**0.797**	0.739	0.717	**0.860**	**0.826**	**0.057**	**0.925**	0.889	0.872	0.949	0.906	0.036	**0.935**	0.908	0.886	**0.942**	**0.914**	0.042	**0.871**	**0.828**	**0.804**	**0.887**	**0.850**	**0.071**
MLMSNet$_{19}$	0.852	0.745	0.761	0.863	0.862	0.049	0.774	0.692	0.681	0.839	0.809	0.064	0.920	0.871	0.860	0.938	**0.907**	0.039	0.928	0.868	0.871	0.916	0.911	0.045	0.864	0.771	0.785	0.847	0.845	0.075
PAGE$_{19}$	0.838	0.777	0.769	0.886	0.854	0.052	0.792	0.736	**0.722**	**0.860**	**0.825**	**0.062**	0.920	0.884	0.868	0.948	0.904	0.036	0.931	0.906	0.886	**0.943**	**0.912**	0.042	0.859	0.817	0.792	0.879	0.840	0.078
HRS$_{19}$	0.843	0.793	0.746	0.889	0.829	0.051	0.762	0.708	0.645	0.842	0.772	0.066	0.913	0.892	0.854	0.938	0.883	0.042	0.920	0.902	0.859	0.923	0.883	0.054	0.852	0.809	0.748	0.850	0.801	0.090
CPD$_{19}$	**0.864**	**0.813**	**0.801**	**0.908**	**0.867**	**0.043**	**0.794**	**0.745**	0.715	**0.868**	0.818	**0.057**	0.924	**0.896**	**0.881**	**0.952**	0.904	**0.033**	**0.936**	**0.914**	**0.894**	**0.943**	0.910	**0.040**	**0.873**	**0.832**	**0.806**	**0.884**	0.843	**0.074**
C2SNet$_{18}$	0.811	0.717	0.717	0.847	0.831	0.062	0.759	0.682	0.663	0.828	0.799	0.072	0.898	0.851	0.834	0.928	0.886	0.047	0.911	0.865	0.854	0.915	0.896	0.053	0.857	0.775	0.777	0.850	0.840	0.080
RAS$_{18}$	0.831	0.751	0.740	0.864	0.839	0.059	0.787	0.713	0.695	0.849	0.814	**0.062**	0.913	0.871	0.843	0.931	0.887	0.045	0.921	0.889	0.857	0.922	0.893	0.056	0.838	0.787	0.738	0.837	0.795	0.104
PAGR$_{18}$	0.854	0.784	0.724	0.883	0.838	0.055	0.771	0.711	0.622	0.843	0.775	0.071	0.919	0.887	0.823	0.941	0.889	0.047	0.927	0.894	0.834	0.917	0.889	0.061	0.858	0.808	0.738	0.854	0.817	0.093
PiCANet$_{18}$	0.851	0.749	0.747	0.865	0.861	0.054	**0.794**	0.710	0.691	0.842	**0.826**	0.068	0.922	0.870	0.848	0.938	0.905	0.042	0.931	0.885	0.865	0.926	**0.914**	0.046	**0.871**	0.804	0.781	0.862	**0.851**	0.077
DSS$^{\dagger}_{17}$	0.825	0.789	0.755	0.885	0.824	0.056	0.781	0.740	0.697	0.844	0.790	**0.063**	0.916	**0.902**	0.867	0.935	0.878	0.040	0.899	0.863	0.822	0.907	0.873	0.068	0.843	0.812	0.762	0.848	0.795	0.096
UCF$_{17}$	0.773	0.631	0.596	0.770	0.782	0.112	0.730	0.621	0.574	0.768	0.760	0.120	0.888	0.823	0.780	0.904	0.874	0.061	0.903	0.844	0.806	0.896	0.884	0.069	0.825	0.738	0.700	0.809	0.807	0.115
MSRNet$_{17}$	0.829	0.723	0.720	0.848	0.839	0.061	0.782	0.687	0.670	0.827	0.808	0.073	0.914	0.866	0.853	0.940	0.903	0.040	0.911	0.868	0.850	0.918	0.895	0.054	0.858	0.790	0.769	0.854	0.841	0.081
NLDF$_{17}$	0.812	0.739	0.710	0.855	0.816	0.065	0.753	0.684	0.634	0.817	0.770	0.080	0.902	0.872	0.839	0.929	0.878	0.048	0.905	0.878	0.839	0.912	0.875	0.063	0.833	0.782	0.742	0.842	0.804	0.099
AMU$_{17}$	0.778	0.678	0.658	0.803	0.804	0.085	0.743	0.647	0.626	0.784	0.781	0.098	0.899	0.843	0.819	0.915	0.886	0.050	0.915	0.868	0.840	0.912	0.894	0.059	0.841	0.771	0.741	0.831	0.821	0.098
ResNet-50/ResNet-101/ResNeXt-101																														
Ours-R	**0.884**	**0.828**	**0.825**	**0.917**	**0.884**	**0.037**	0.810	**0.756**	**0.738**	**0.873**	0.833	**0.055**	**0.935**	**0.908**	**0.899**	**0.961**	**0.920**	**0.028**	**0.947**	**0.924**	**0.911**	**0.953**	**0.925**	**0.033**	**0.882**	**0.842**	**0.821**	**0.899**	**0.857**	**0.064**
SCRN$_{19}$	**0.888**	**0.809**	0.803	0.901	**0.885**	**0.040**	**0.811**	0.746	0.720	**0.869**	**0.837**	**0.056**	**0.935**	0.897	0.878	0.954	**0.917**	0.033	**0.950**	0.918	0.899	0.942	**0.927**	**0.037**	**0.890**	**0.839**	**0.816**	**0.888**	**0.867**	**0.065**
EGNet-R$_{19}$	**0.889**	**0.815**	**0.816**	**0.907**	**0.887**	**0.039**	**0.815**	**0.756**	**0.738**	**0.874**	**0.841**	**0.053**	**0.935**	**0.901**	**0.887**	**0.956**	**0.918**	**0.031**	**0.947**	**0.920**	0.903	**0.947**	**0.925**	**0.037**	0.878	0.831	0.807	0.879	0.853	0.075
CPD-R$_{19}$	0.865	0.805	0.795	**0.904**	0.869	0.043	0.797	**0.747**	0.719	**0.873**	0.825	**0.056**	**0.925**	0.891	0.876	0.952	0.906	0.034	0.939	0.917	0.898	**0.949**	0.918	**0.037**	0.872	0.831	0.803	0.887	0.847	0.072
ICNet$_{19}$	0.855	0.767	0.762	0.880	0.865	0.048	**0.813**	0.739	0.730	0.859	**0.837**	0.061	**0.925**	0.880	0.858	0.943	0.908	0.037	0.938	0.880	0.881	0.923	0.918	0.041	0.866	0.786	0.790	0.860	0.850	0.071
BANet$_{19}$	0.872	**0.815**	**0.811**	**0.907**	0.879	**0.040**	0.803	0.746	**0.736**	0.865	0.832	0.059	**0.930**	**0.899**	**0.887**	**0.955**	0.913	**0.032**	**0.945**	**0.923**	**0.908**	**0.953**	**0.924**	**0.035**	**0.879**	**0.838**	**0.817**	**0.889**	0.853	**0.070**
BASNet$_{19}$	0.859	0.791	0.803	0.884	0.866	0.048	0.805	**0.756**	**0.751**	**0.869**	**0.836**	**0.056**	**0.930**	0.898	**0.890**	0.947	0.908	0.033	0.942	0.879	**0.904**	0.921	0.916	**0.037**	0.863	0.781	0.800	0.853	0.837	0.077
CapSal$^{R101}_{19}$	0.823	0.755	0.691	0.866	0.815	0.062	0.639	0.564	0.484	0.703	0.674	0.096	0.884	0.843	0.782	0.907	0.850	0.058	0.862	0.825	0.771	0.866	0.826	0.074	0.869	0.827	0.791	0.878	0.837	0.074
R3Net$^{*X101}_{18}$	0.833	0.787	0.767	0.879	0.836	0.057	0.795	**0.748**	0.728	0.859	0.817	0.062	0.915	0.894	0.878	0.945	0.895	0.035	0.934	0.914	0.902	0.940	0.910	0.040	0.846	0.805	0.765	0.846	0.805	0.094
DGRL$_{18}$	0.828	0.794	0.774	0.899	0.842	0.050	0.779	0.709	0.697	0.850	0.810	0.063	0.914	0.882	0.865	0.947	0.896	0.038	0.925	0.903	0.883	0.943	0.906	0.043	0.860	0.814	0.792	0.881	0.839	0.075
PiCANet-R$_{18}$	0.860	0.759	0.755	0.873	0.869	0.051	0.803	0.717	0.695	0.848	0.832	0.065	0.919	0.870	0.842	0.941	0.905	0.044	0.935	0.886	0.867	0.927	0.917	0.046	0.870	0.804	0.782	0.862	**0.854**	0.076
SRM$_{17}$	0.826	0.753	0.722	0.867	0.836	0.059	0.769	0.707	0.658	0.843	0.798	0.069	0.906	0.873	0.835	0.939	0.887	0.046	0.917	0.892	0.853	0.928	0.895	0.054	0.850	0.804	0.762	0.861	0.833	0.085

where λ is a hyperparameter that balances the contributions of the two losses. For the sake of simplicity, it is set to 1.

4. Experiments

4.1. Datasets

We evaluate the proposed model on five benchmark datasets: DUTS [31], DUT-OMRON [45], ECSSD [44], HKU-IS [16], and PASCAL-S [18]. The DUTS contains 10,553 training and 5,019 test images, which is currently the largest salient object detection dataset. Both training and test sets contain complicated scenes. The DUT-OMRON contains 5,168 images of complex backgrounds and high content variety. The ECSSD is composed of 1,000 images with structurally complex natural contents. The HKU-IS contains 4,447 complex scenes that contain multiple disconnected objects with relatively diverse spatial distributions, and a similar fore-/back-ground appearance makes it more difficult to distinguish. We follow the data partition of [16, 13, 4] to use 1,447 images for testing. The PASCAL-S consists of 850 challenging images.

4.2. Evaluation Criteria

In this paper, we use six measurements to eveluate every models. **Precision-Recall (PR) curve.** We binarize the gray-scale prediction by a fixed threshold. The resulted binary map and the ground truth are used to calculate Precision $= \mathrm{TP}/(\mathrm{TP}+\mathrm{FP})$ and Recall $= \mathrm{TP}/(\mathrm{TP}+\mathrm{FN})$, where TP, FP and FN represent true-positive, false-positive and false-negative, respectively. The PR curve can be plotted by a group of pairs of precision and recall generated when the threshold slides from 0 to 255. The larger the area under the PR curve, the better the performance. **F-measure [1], $\mathbf{F}_\beta$** is formulated as the weighted harmonic mean of Precision and Recall [1], e.t. $\mathrm{F}_\beta = \frac{(1+\beta^2)\mathrm{Precision}\times\mathrm{Recall}}{\beta^2\mathrm{Precision}+\mathrm{Recall}}$, where β^2 is generally set to 0.3 to emphasize more on the precision. We calculate the maximal F_β values from the PR curve, denoted as F_{max} and use an adaptive threshold that is twice the mean value of the prediction to calculate F_{avg}. And the F_{avg} can reflect the spatial consistency of the predictions [41]. **MAE [26]** directly evaluates the average pixel-level relative error between the normalized prediction and the ground truth by calculating the mean of the absolute value of the difference. **S-measure [7], $\mathbf{S}_m$** computes the object-aware and region-aware structure similarities, denoted as S_o and S_r, between the prediction and the ground truth. S-measure is written as follows: $\mathrm{S}_m = \alpha\cdot\mathrm{S}_o + (1-\alpha)\cdot\mathrm{S}_r$, where α is set to 0.5 [7]. **E-measure [8], $\mathbf{E}_m$** combines local pixel values with the image-level mean value to jointly evaluate the similarity between the prediction and the ground truth. **weighted F-measure [25], $\mathbf{F}^{\omega}_{\beta}$** defines a weighted precision, which is a measure of exactness, and a weighted recall, which is a measure of completeness is proposed to improve the existing metric to improve the existing metric F-measure.

4.3. Implementation Details

Following most existing state-of-the-art methods [30, 27, 41, 34, 37, 46, 48, 20, 53, 32, 42, 56], we use the DUTS-

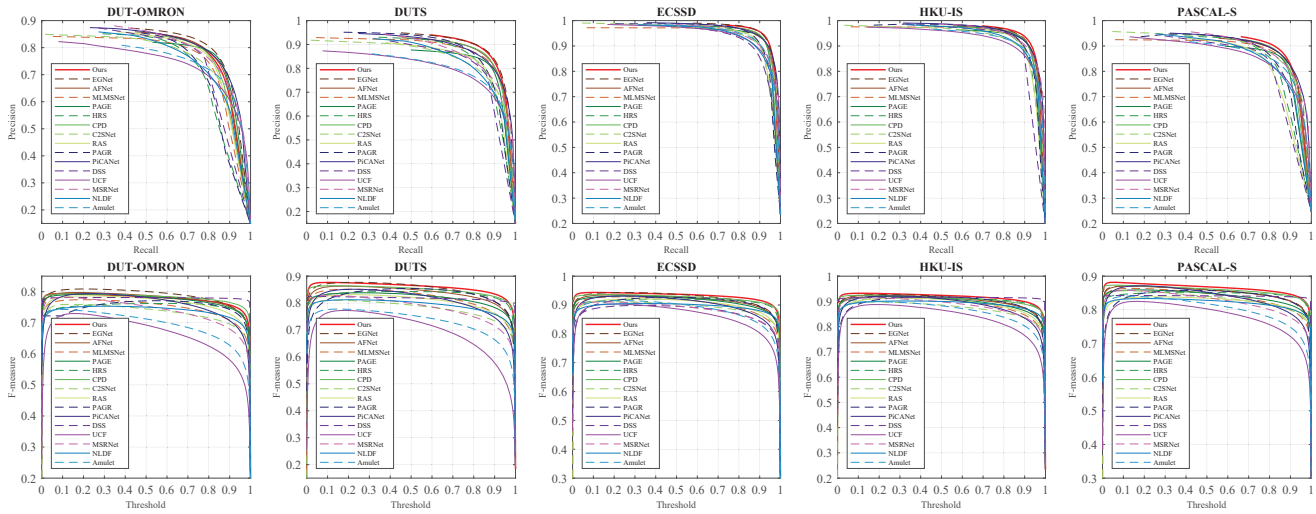

Figure 6. Precision-Recall curves (1^{st} row) and F-measure curves (2^{nd} row) on five common saliency datasets.

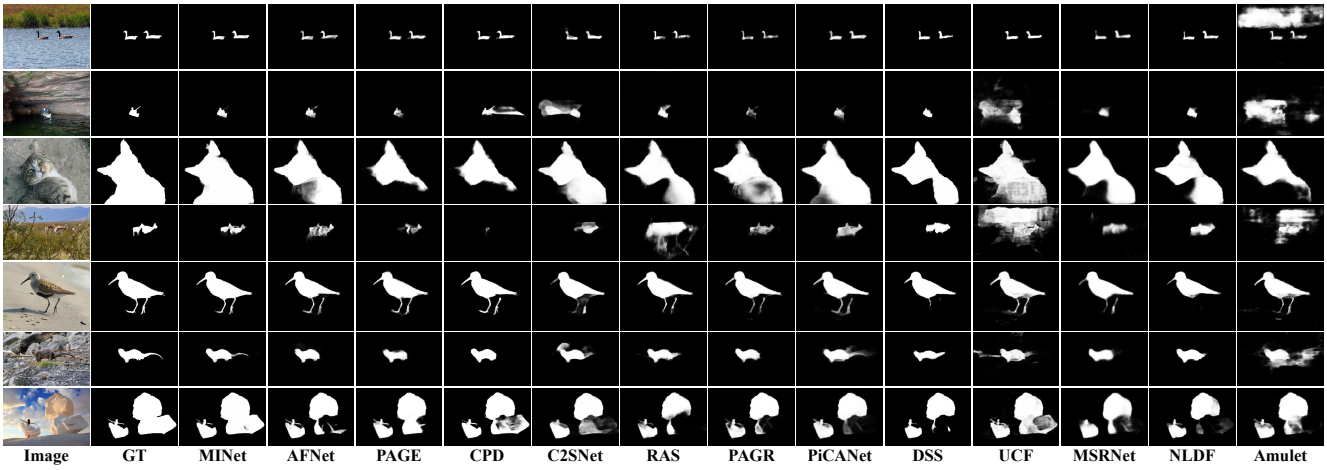

Figure 7. Visual comparisons of different methods.

TR [31] as the training dataset. During the training stage, random horizontal flipping, random rotating, and random color jittering act as data augmentation techniques to avoid the over-fitting problem. To ensure model convergence, our network is trained for 50 epochs with a mini-batch of 4 on an NVIDIA GTX 1080 Ti GPU. The backbone parameters (i.e. VGG-16 and ResNet-50) are initialized with the corresponding models pretrained on the ImageNet dataset and the rest ones are initialized by the default setting of PyTorch. We use the momentum SGD optimizer with a weight decay of 5e-4, an initial learning rate of 1e-3 and a momentum of 0.9. Moreover, we apply a "poly" strategy [21] with a factor of 0.9. The input size is 320×320.

4.4. Comparison with State-of-the-arts

We compare the proposed algorithm with 23 state-of-the-art saliency detection methods, including the SRM [32], PiCANet [20], DGRL [34], R3Net [6], CapSal [50], BASNet [27], BANet [30], ICNet [36], CPD [41], Amulet [51], NLDF [23], MSRNet [15], UCF [52], DSS [13], PAGR[53], RAS [4], C2SNet [17], HRS [46], PAGE [37], MLMSNet [40], AFNet [9], SCRN [42], and EGNet [56]. For fair comparisons, all saliency maps of these methods are provided by authors or computed by their released codes.

Quantitative Comparison. To fully compare the proposed method with these existing models, the detailed experimental results in terms of six metrics are listed in Tab. 1. As can be seen from the results, our approach has shown very good performance and significantly outperforms other competitors, although some methods [13, 6] use CRF [14] or other post-processing methods. The proposed method consistently performs better than all the competitors across all six metrics on most datasets. In particular, in terms of the MAE, the performance is averagely improved by 8.11%

Table 2. Ablation analysis on the DUTS-TE dataset.

Model	F_{max}	F_{avg}	F_β^ω	E_m	S_m	MAE
Baseline	0.829	0.738	0.725	0.859	0.842	0.057
+AIMs	0.855	0.775	0.768	0.884	0.860	0.047
+Amulet-like	0.845	0.758	0.747	0.872	0.851	0.052
+SIMs	0.865	0.786	0.773	0.888	0.865	0.047
+PPM	0.847	0.762	0.753	0.875	0.856	0.050
+ASPP	0.859	0.777	0.767	0.880	0.861	0.048
+AIMs+SIMs	0.874	0.792	0.789	0.893	0.874	0.044
+AIMs+SIMs+CEL	0.877	0.823	0.813	0.912	0.875	0.039

and 7.30% over the second-best method CPD [41] with the VGG-16 as the backbone and EGNet [56] with the ResNet-50 as the backbone, respectively. In addition, we demonstrate the standard PR curves and the F-measure curves in Fig. 6. Our approach (red solid line) achieves the best results on the DUTS-TE, ECSSD, PASCAL-S and HKU-IS datasets and is also very competitive on the DUT-OMRON.

Qualitative Evaluation. Some representative examples are shown in Fig. 7. These examples reflect various scenarios, including small objects (1^{st} and 2^{nd} rows), low contrast between salient object and image background (3^{rd} and 4^{th} rows), objects with threadlike parts (5^{th} and 6^{th} rows) and complex scene (6^{th} and 7^{th} rows). Moreover, these images contain small-/middle- and large-scale objects. It can be seen that the proposed method can consistently produce more accurate and complete saliency maps with sharp boundaries and coherent details.

4.5. Ablation Study

To illustrate the effectiveness of each proposed module, we conduct a detailed analysis next.

Effectiveness of the AIMs and SIMs. Our baseline model is an FPN-like network [19], which uses the lateral connections to reduce the channel number to 32 in the shallowest layer and to 64 in the other layers. We separately install the AIMs and SIMs on the baseline network and evaluate their performance. The results are shown in Tab. 2. It can be seen that both modules achieve significant performance improvement over the baseline. And, the proposed SIMs also performs much better than the PPM [55] and the ASPP [3] and it has increased by 6.21% and 1.45% in MAE, especially. In addition, the combination of the AIMs and SIMs can further improve the performance. The visual effects of different modules are illustrated in Fig. 8. We can see that the AIMs and SIMs help effectively suppress the interference of backgrounds and completely segment salient objects because the richer multi-scale contextual information can be captured by the interactive feature learning.

Comparisons with the Amulet-like [51] strategy. We compare the AIMs with the Amulet-like strategy in FLOPs, Parameters and GPU memory. "+AIMs": 137G, 47M and 1061MiB. "+Amulet-like": 176G, 20M and 1587MiB. AIMs combine fewer levels and have less computational cost. The fusion strategy achieves higher accuracy. And in Tab. 2, it gets additional 2.14%, 2.77% and 8.26% improvement in F_{avg}, F_β^ω and MAE over the model "+Amulet-like".

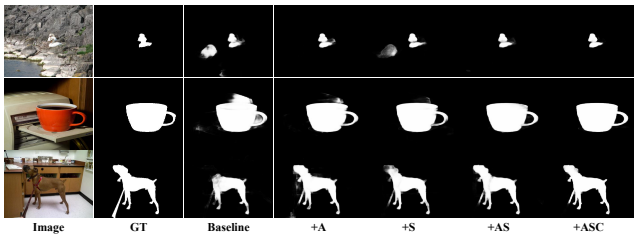

Figure 8. Visual comparisons for showing the benefits of the proposed modules. GT: Ground truth; A: AIMs; S: SIMs; C: CEL.

Effectiveness of the CEL. We also quantitatively evaluate the effect of the consistency-enhanced loss (CEL) in Tab. 2. Compared to "+AIMs+SIMs", the model with the CEL achieves consistent performance enhancements in terms of all six metrics. In particular, the F_{avg}, F_β^ω and MAE scores are respectively improved by 4.75%, 3.75%, and 13.16%. Since the F_{avg} is closely related to the spatial consistency of the predicted results [41], the salient regions are more uniformly highlighted as shown in Fig. 8.

5. Conclusion

In this paper, we investigate the multi-scale issue to propose an effective and efficient network MINet with the transformation-interaction-fusion strategy, for salient object detection. We first use the aggregate interaction modules (AIMs) to integrate the similar resolution features of adjacent levels in the encoder. Then, the self-interaction modules (SIMs) are utilized to extract the multi-scale information from a single level feature for the decoder. Both AIMs and SIMs interactively learn contextual knowledge from the branches of different resolutions to boost the representation capability of size-varying objects. Finally, we employ the consistency-enhanced loss (CEL) to alleviate the fore- and back-ground imbalance issue, which can also help uniformly highlight salient object regions. Each proposed module achieves significant performance improvement. Extensive experiments on five datasets validate that the proposed model outperforms 23 state-of-the-art methods under different evaluation metrics.

Acknowledgements

This work was supported in part by the National Key R&D Program of China #2018AAA0102003, National Natural Science Foundation of China #61876202, #61725202, #61751212 and #61829102, and the Dalian Science and Technology Innovation Foundation #2019J12GX039.

References

[1] Radhakrishna Achanta, Sheila Hemami, Francisco Estrada, and Sabine Süsstrunk. Frequency-tuned salient region detection. In *CVPR*, number CONF, pages 1597–1604, 2009.

[2] Ali Borji, Ming-Ming Cheng, Qibin Hou, Huaizu Jiang, and Jia Li. Salient object detection: A survey. *Computational Visual Media*, 5:117–150, 2014.

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE TPAMI*, 40(4):834–848, 2017.

[4] Shuhan Chen, Xiuli Tan, Ben Wang, and Xuelong Hu. Reverse attention for salient object detection. In *ECCV*, pages 234–250, 2018.

[5] Ming-Ming Cheng, Niloy J Mitra, Xiaolei Huang, Philip HS Torr, and Shi-Min Hu. Global contrast based salient region detection. *IEEE TPAMI*, 37(3):569–582, 2014.

[6] Zijun Deng, Xiaowei Hu, Lei Zhu, Xuemiao Xu, Jing Qin, Guoqiang Han, and Pheng-Ann Heng. R3net: Recurrent residual refinement network for saliency detection. In *IJCAI*, pages 684–690, 2018.

[7] Deng-Ping Fan, Ming-Ming Cheng, Yun Liu, Tao Li, and Ali Borji. Structure-measure: A new way to evaluate foreground maps. In *ICCV*, pages 4548–4557, 2017.

[8] Deng-Ping Fan, Cheng Gong, Yang Cao, Bo Ren, Ming-Ming Cheng, and Ali Borji. Enhanced-alignment measure for binary foreground map evaluation. In *IJCAI*, pages 698–704, 2018.

[9] Mengyang Feng, Huchuan Lu, and Errui Ding. Attentive feedback network for boundary-aware salient object detection. In *CVPR*, pages 1623–1632, 2019.

[10] Yue Gao, Meng Wang, Dacheng Tao, Rongrong Ji, and Qionghai Dai. 3-d object retrieval and recognition with hypergraph analysis. *IEEE TIP*, 21(9):4290–4303, 2012.

[11] Fang Guo, Wenguan Wang, Jianbing Shen, Ling Shao, Jian Yang, Dacheng Tao, and Yuan Yan Tang. Video saliency detection using object proposals. *IEEE Transactions on Cybernetics*, 48(11):3159–3170, 2017.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[13] Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, and Philip HS Torr. Deeply supervised salient object detection with short connections. In *CVPR*, pages 3203–3212, 2017.

[14] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NEURIPS*, pages 109–117, 2011.

[15] Guanbin Li, Yuan Xie, Liang Lin, and Yizhou Yu. Instance-level salient object segmentation. In *CVPR*, pages 2386–2395, 2017.

[16] Guanbin Li and Yizhou Yu. Visual saliency based on multiscale deep features. In *CVPR*, pages 5455–5463, 2015.

[17] Xin Li, Fan Yang, Hong Cheng, Wei Liu, and Dinggang Shen. Contour knowledge transfer for salient object detection. In *ECCV*, pages 355–370, 2018.

[18] Yin Li, Xiaodi Hou, Christof Koch, James M Rehg, and Alan L Yuille. The secrets of salient object segmentation. In *CVPR*, pages 280–287, 2014.

[19] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 2117–2125, 2017.

[20] Nian Liu, Junwei Han, and Ming-Hsuan Yang. Picanet: Learning pixel-wise contextual attention for saliency detection. In *CVPR*, pages 3089–3098, 2018.

[21] Wei Liu, Andrew Rabinovich, and Alexander C Berg. Parsenet: Looking wider to see better. *arXiv preprint arXiv:1506.04579*, 2015.

[22] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015.

[23] Zhiming Luo, Akshaya Mishra, Andrew Achkar, Justin Eichel, Shaozi Li, and Pierre-Marc Jodoin. Non-local deep features for salient object detection. In *CVPR*, pages 6609–6617, 2017.

[24] Vijay Mahadevan and Nuno Vasconcelos. Saliency-based discriminant tracking. In *CVPR*, 2009.

[25] Ran Margolin, Lihi Zelnik-Manor, and Ayellet Tal. How to evaluate foreground maps? In *CVPR*, pages 248–255, 2014.

[26] Federico Perazzi, Philipp Krähenbühl, Yael Pritch, and Alexander Hornung. Saliency filters: Contrast based filtering for salient region detection. In *CVPR*, pages 733–740, 2012.

[27] Xuebin Qin, Zichen Zhang, Chenyang Huang, Chao Gao, Masood Dehghan, and Martin Jagersand. Basnet: Boundary-aware salient object detection. In *CVPR*, pages 7479–7489, 2019.

[28] Paul L Rosin and Yu-Kun Lai. Artistic minimal rendering with lines and blocks. *Graphical Models*, 75(4):208–229, 2013.

[29] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[30] Jinming Su, Jia Li, Changqun Xia, and Yonghong Tian. Selectivity or invariance: Boundary-aware salient object detection. *arXiv preprint arXiv:1812.10066*, 2018.

[31] Lijun Wang, Huchuan Lu, Yifan Wang, Mengyang Feng, Dong Wang, Baocai Yin, and Xiang Ruan. Learning to detect salient objects with image-level supervision. In *CVPR*, pages 136–145, 2017.

[32] Tiantian Wang, Ali Borji, Lihe Zhang, Pingping Zhang, and Huchuan Lu. A stagewise refinement model for detecting salient objects in images. In *ICCV*, pages 4019–4028, 2017.

[33] Tiantian Wang, Yongri Piao, Xiao Li, Lihe Zhang, and Huchuan Lu. Deep learning for light field saliency detection. In *ICCV*, pages 8838–8848, 2019.

[34] Tiantian Wang, Lihe Zhang, Shuo Wang, Huchuan Lu, Gang Yang, Xiang Ruan, and Ali Borji. Detect globally, refine locally: A novel approach to saliency detection. In *CVPR*, pages 3127–3135, 2018.

[35] Wenguan Wang, Qiuxia Lai, Huazhu Fu, Jianbing Shen, and Haibin Ling. Salient object detection in the deep learning era: An in-depth survey. *arXiv preprint arXiv:1904.09146*, 2019.

[36] Wenguan Wang, Jianbing Shen, Ming-Ming Cheng, and Ling Shao. An iterative and cooperative top-down and bottom-up inference network for salient object detection. In *CVPR*, pages 5968–5977, 2019.

[37] Wenguan Wang, Shuyang Zhao, Jianbing Shen, Steven CH Hoi, and Ali Borji. Salient object detection with pyramid attention and salient edges. In *CVPR*, pages 1448–1457, 2019.

[38] Xiaochuan Wang, Xiaohui Liang, Bailin Yang, and Frederick WB Li. No-reference synthetic image quality assessment with convolutional neural network and local image saliency. *Computational Visual Media*, 5(2):193–208, 2019.

[39] Yichen Wei, Fang Wen, Wangjiang Zhu, and Jian Sun. Geodesic saliency using background priors. In *ECCV*, pages 29–42, 2012.

[40] Runmin Wu, Mengyang Feng, Wenlong Guan, Dong Wang, Huchuan Lu, and Errui Ding. A mutual learning method for salient object detection with intertwined multi-supervision. In *CVPR*, pages 8150–8159, 2019.

[41] Zhe Wu, Li Su, and Qingming Huang. Cascaded partial decoder for fast and accurate salient object detection. In *CVPR*, pages 3907–3916, 2019.

[42] Zhe Wu, Li Su, and Qingming Huang. Stacked cross refinement network for edge-aware salient object detection. In *ICCV*, pages 7264–7273, 2019.

[43] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, pages 1492–1500, 2017.

[44] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Hierarchical saliency detection. In *CVPR*, pages 1155–1162, 2013.

[45] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *CVPR*, pages 3166–3173, 2013.

[46] Yi Zeng, Pingping Zhang, Jianming Zhang, Zhe Lin, and Huchuan Lu. Towards high-resolution salient object detection. *arXiv preprint arXiv:1908.07274*, 2019.

[47] Lihe Zhang, Jianwu Ai, Bowen Jiang, Huchuan Lu, and Xiukui Li. Saliency detection via absorbing markov chain with learnt transition probability. *IEEE TIP*, 27(2):987–998, 2018.

[48] Lu Zhang, Ju Dai, Huchuan Lu, You He, and Gang Wang. A bi-directional message passing model for salient object detection. In *CVPR*, pages 1741–1750, 2018.

[49] Lihe Zhang, Chuan Yang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Ranking saliency. *IEEE TPAMI*, 39(9):1892–1904, 2017.

[50] Lu Zhang, Jianming Zhang, Zhe Lin, Huchuan Lu, and You He. Capsal: Leveraging captioning to boost semantics for salient object detection. In *CVPR*, pages 6024–6033, 2019.

[51] Pingping Zhang, Dong Wang, Huchuan Lu, Hongyu Wang, and Xiang Ruan. Amulet: Aggregating multi-level convolutional features for salient object detection. In *ICCV*, pages 202–211, 2017.

[52] Pingping Zhang, Dong Wang, Huchuan Lu, Hongyu Wang, and Baocai Yin. Learning uncertain convolutional features for accurate saliency detection. In *ICCV*, pages 212–221, 2017.

[53] Xiaoning Zhang, Tiantian Wang, Jinqing Qi, Huchuan Lu, and Gang Wang. Progressive attention guided recurrent network for salient object detection. In *CVPR*, pages 714–722, 2018.

[54] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *CVPR*, pages 4320–4328, 2018.

[55] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, pages 2881–2890, 2017.

[56] Jia-Xing Zhao, Jiang-Jiang Liu, Deng-Ping Fan, Yang Cao, Jufeng Yang, and Ming-Ming Cheng. Egnet: Edge guidance network for salient object detection. In *ICCV*.

[57] Rui Zhao, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Saliency detection by multi-context deep learning. In *CVPR*, pages 1265–1274, 2015.