Liu *et al*. *Intell Robot* 2024;4(4):503-23
DOI: 10.20517/ir.2024.29

Intelligence & Robotics

Research Article

Open Access

SANet: scale-adaptive network for lightweight salient object detection

Zhuang Liu, Weidong Zhao, Ning Jia, Xianhui Liu, Jiaxiong Yang

College of Electronics and Information Engineering, Tongji University, Shanghai 201804, China.

Correspondence to: Prof. Weidong Zhao, College of Electronics and Information Engineering, Tongji University, No. 4800, Cao'an Highway, Jiading District, Shanghai 201804, China. E-mail: zhaowd494@163.com

How to cite this article: Liu Z, Zhao W, Jia N, Liu X, Yang J. SANet: scale-adaptive network for lightweight salient object detection. *Intell Robot* 2024;4(4):503-23. http://dx.doi.org/10.20517/ir.2024.29

Received: 18 Sep 2024 **First Decision:** 19 Nov 2024 **Revised:** 17 Dec 2024 **Accepted:** 19 Dec 2024 **Published:** 31 Dec 2024

Academic Editor: Simon Yang **Copy Editor:** Pei-Yun Wang **Production Editor:** Pei-Yun Wang

Abstract

Salient object detection (SOD) is widely used in transportation such as road damage detection, assisted driving, *etc*. However, heavyweight SOD methods are difficult to apply in scenarios with low computing power due to their huge amount of computation and parameters. The detection accuracy of most lightweight SOD methods is difficult to meet application requirements. We propose a novel lightweight scale-adaptive network to achieve a trade-off between lightweight restriction and detection performance. We first propose the scale-adaptive feature extraction (SAFE) module, which mainly consists of two parts: multi-scale feature interaction, which can extract features of different scales and enhance the representation ability of the network; and dynamic selection, which can adaptively assign different weights to features of varying scales according to their contribution through the input image. Then, based on the SAFE module, a lightweight and adaptive backbone network is designed, and scale-adaptive network is implemented in combination with the multi-scale feature aggregation (MFA) module. We evaluate the model quantitatively and qualitatively on six public datasets and compare it with typical heavyweight and lightweight methods. With only 2.29 M parameters, it can achieve a prediction speed of 62 fps on a GTX 3090 GPU, far exceeding other models, and real-time performance is guaranteed. The model performance reaches that of general heavyweight methods and exceeds state-of-the-art lightweight methods.

Keywords: Salient object detection, lightweight SOD, model lightweighting, multi-scale learning

1. INTRODUCTION

Salient object detection (SOD) aims to detect the most distinctive objects in natural images[1]. The initial SOD model was inspired by cognitive psychology and neuroscience, proposed by Itti *et al*. in 1998[2]. Different from traditional methods, Liu *et al*. formulated SOD as a binary labeling problem to separate salient objects from the background and proposed a set of new features, including multi-scale contrast, center-surround histogram, and color space distribution to describe local, regional, and global salient objects[3]. They also built the first large-scale image database for the quantitative evaluation of visual attention algorithms that many inspired researchers began to propose more SOD models. SOD can be used in many fields such as object detection[4], person re-identification[5], and especially in transportation. As shown in Figure 1, SOD is widely used in road damage detection[6], assisted driving[7–9], *etc.* In autonomous driving vision systems, SOD can quickly allocate attention to important objects for scene analysis[10,11]. However, heavyweight SOD methods are difficult to apply in industrial scenarios with low computing power due to their huge amount of computation and parameters. In the field of autonomous driving or assisted driving, the onboard computer will process all objects in the traffic scene indiscriminately. This reduces the efficiency of information processing and prolongs the processing time of some emergencies[12]. In some special scenarios, sometimes only some special objects need to be detected, such as vehicles in front, traffic signs, pedestrians on the roadside, *etc.* This is precisely the advantage of SOD. However, there are still the following difficulties in applying SOD in the field of intelligent transportation: (1) Since all objects that affect driving should be regarded as salient targets, there will not be only one salient target in most driving scenes, which puts higher requirements on the model; (2) Traffic scenes are extremely complex, and the general SOD model cannot achieve good results; (3) Traffic scenes require a higher model processing speed, and the existing SOD model cannot meet the requirements. How to design and implement a SOD model that considers both real-time and detection performance remains a critical challenge.

Traditional SOD methods mainly rely on low-level image features and heuristic priors, but the lack of guidance from high-level semantic information usually leads to limited accuracy. In recent years, with the rise of convolutional neural networks (CNNs), especially fully convolutional networks (FCNs), deep learning-based methods have pushed SOD to a new level. However, these outstanding performances are often achieved at the expense of high computing costs and demanding software and hardware requirements[13]. For example, multi-scale interactive network (MINet)[14] with VGG-16 backbone contains 162.38 M parameters, and the floating-point operations (FLOPs) reach 87.1 G. Although it demonstrates good detection performance, it cannot be deployed in low computing power environments. Therefore, it is necessary to design a lightweight SOD method with excellent performance to serve actual application scenarios.

Cross-stage cross-scale network (CSNet)[15], hierarchical visual perception module-incorporated lightweight SOD network (HVPNet)[16], and stereoscopically attentive multi-scale network (SAMNet)[17] are three representative lightweight SOD methods. CSNet is designed to be lightweight based on the dynamic weight decay pruning method, while HVPNet and SAMNet achieve model lightweighting by improving the network structure. Compared with MINet, the parameters of CSNet, HVPNet and SAMNet are only 0.14, 1.24, and 1.33 M, respectively. However, it is worth noting that although these models are lightweight enough, their detection effect is poor, as shown in Figure 2, making them difficult to apply in some complex scenarios. Therefore, realizing a SOD model that considers both lightweight and detection effect is a very challenging task. The main difficulties this work faces are as follows: (1) The lightweight network has a simple structure and can process a small feature domain, which cannot comprehensively represent salient objects. Simply using existing lightweight backbone networks (MobileNet[18,19] or ShuffleNet[20,21], *etc.*) directly for SOD tasks does not produce ideal results, which will be demonstrated in the experiments; (2) In complex scenes, salient objects are scale-variable. How to make the model adaptively and dynamically perceive and extract the features of salient objects is another difficult problem we need to deal with; (3) Current mainstream lightweight SOD methods cannot simultaneously achieve both lightweight design and high performance. Properly balancing these two

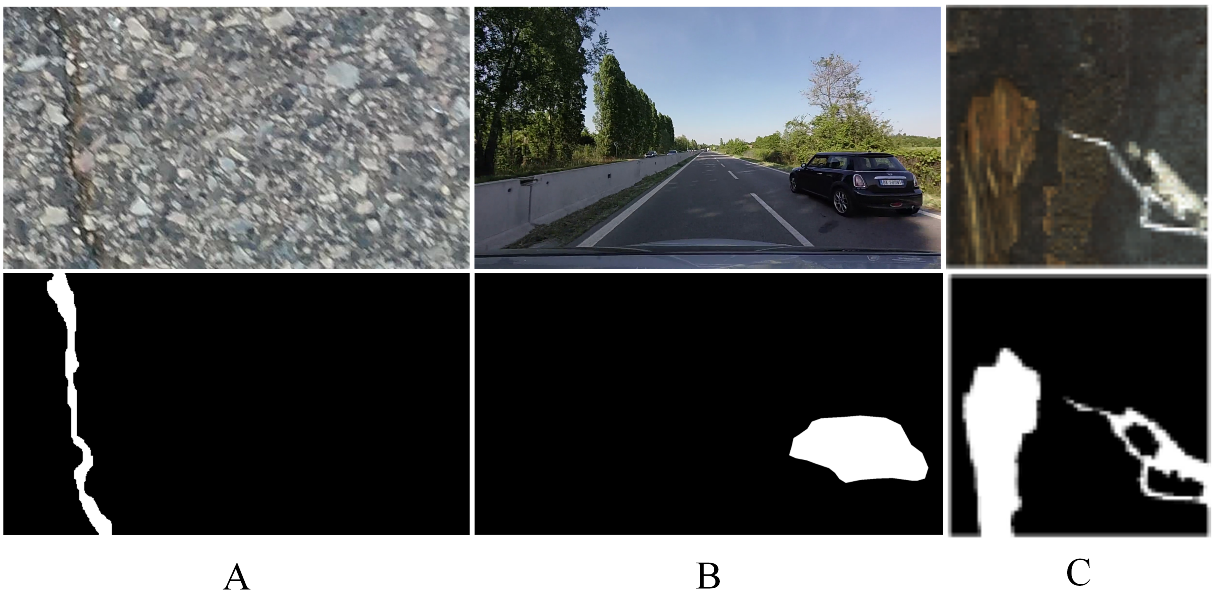

Figure 1. Application scenarios of SOD. (A) Road surface defect detection; (B) Assisted driving; (C) Strip steel surface defect detection. SOD: Salient object detection.

aspects remains a challenge in SOD.

We design and implement an efficient and lightweight SOD model based on the above analysis. It adopts a novel scale-adaptive feature extraction (SAFE) module for multi-scale learning. Meanwhile, it can adaptively adjust the weight of each scale of information according to its importance to achieve dynamic perception of the features of salient objects. The SAFE module mainly consists of multi-scale feature interaction and dynamic selection. The multi-scale feature interaction is mainly used for feature extraction. It first uses depthwise separable convolutions with different dilation rates to extract information of various receptive fields, and divides the input features into distinct branches. Then, feature interaction is achieved by fusing the features of different branches to improve their representation capabilities. The dynamic selection mainly combines channel attention with multi-layer perceptron (MLP) to assign different weights to features of multiple scales to extract key feature information. We also designed a decoder based on the multi-scale feature aggregation (MFA) module to alleviate the information loss problem caused by excessive upsampling. Based on the SAFE and MFA modules, we implement an encoder-decoder network that is more suitable for SOD tasks, namely scale-adaptive network (SANet). It can achieve an inference speed [frames per second (FPS)] of 62 fps on an NVIDIA GTX 3090 GPU with only 2.29 M parameters, far exceeding other models, and real-time performance is guaranteed. The model performance reaches that of general heavyweight methods and exceeds many first-class lightweight methods.

In summary, our contributions mainly include the following three points:

(1) We propose a novel SAFE module, which consists of two parts: multi-scale feature interaction, which is used to extract features of different scales and enhance the representation of salient objects through the interaction of cross-scale features; dynamic selection, which is data-driven and can adaptively perceive and measure the importance of features of different scales according to the changes in the input images.

(2) We implement the SANet network, which consists of an encoder based on the SAFE module and a decoder based on the MFA module. This is an encoder-decoder network that considers both lightweight and detection

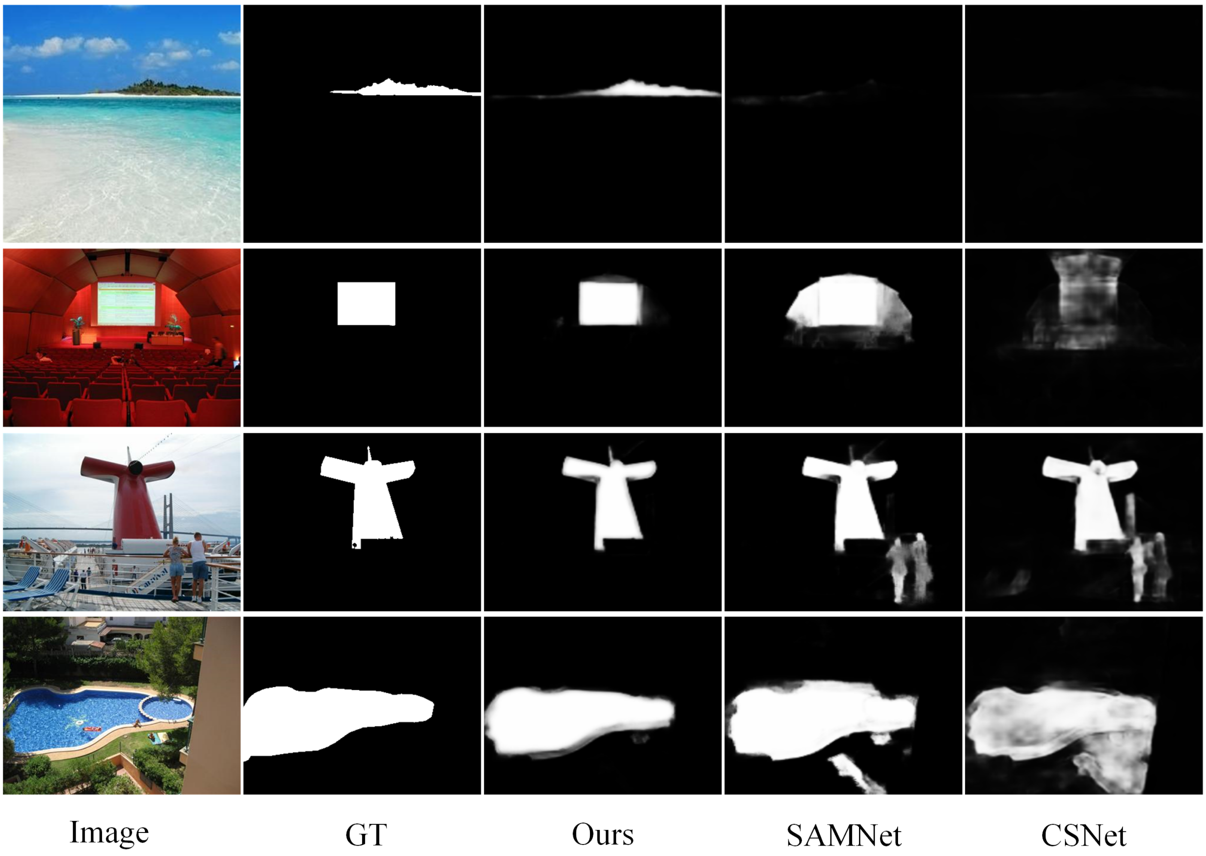

Figure 2. Comparison of detection effect of different methods.

performance.

(3) We quantitatively and qualitatively compare SANet with fifteen heavyweight methods and three lightweight methods on six typical SOD datasets. At the same time, we use the traffic dataset traffic salient object detection (TSOD) mentioned in TSOD using a feature deep interaction and guidance fusion network (TFGNet)[22] to verify our model. SANet demonstrates excellent detection effect and efficient reasoning speed with low parameters and model complexity.

2. RELATED WORKS

2.1. SOD

SOD is based on simulating a human visual attention mechanism, which enables machines to automatically discover and filter important information. Since Professor Itti pioneered the research field of SOD in 1999, countless researchers have been engaged in research in this field and produced many scientific research results. SOD's technical solutions have also shifted from traditional statistical methods, frequency domain conversion methods, and machine learning methods to the currently hot field of deep learning. Traditional methods[23] are mainly based on manually designed features. Although they are very efficient, manually designed features inherently lack the ability of high-level representation, which limits the performance of the model. The deep learning-based methods have shown incomparable advantages over traditional methods in characterizing salient objects. They have quickly occupied the forefront of SOD and raised the level of SOD to a new height. Early deep learning-based methods did not solve the problem of longitudinal transmission feature attenuation, and the model had problems of false positives or negatives[24]. To this end, the encoder network is applied to

SOD. Refs.[14] and[25] also designed a feature conversion module to improve the effectiveness of horizontal feature transmission. To embed semantic information into the encoding and decoding processes, Chen *et al*. and Jia *et al*. designed different global information enhancement modules respectively[1,26]. The Transformer model has facilitated a further enhancement in the level of SOD. However, the early transformer-based SOD models[27] were relatively complex and were not very suitable for high-resolution SOD tasks. Although some lightweight transformer networks[28] have been proposed in recent years and have reduced the number of model parameters from level B to around 80 M, this is still not affordable for edge applications. Current SOD methods based on large models still have the problem of high model complexity.

In summary, relevant research on SOD has accumulated a lot of research results, and the detection effect has reached the level of practical application. However, this is achieved under ideal laboratory conditions. The complexity and real-time performance of the model cannot meet the requirements in weak computing and high real-time scenarios.

2.2. Model lightweighting

Lightweight models have attracted attention in various fields due to their low computing resource requirements. There are two main methods to build lightweight models. One is to use network pruning, model quantization, or knowledge distillation to make complex models lightweight. Network pruning reduces the size of a neural network by removing unnecessary connections or nodes[29]. Model quantization reduces the storage space and computing resources required by the model by reducing the number of bits of the parameters and representing the parameters as integers or fixed-point numbers with fewer bits[30]. The knowledge distillation method achieves model lightweighting by transferring knowledge between large models and small models[31]. The other approach is to consider lightweight from the network design stage, to design an efficient and lightweight backbone network. Lightweight network design has been a research hotspot in the field of deep learning in recent years, aiming to provide efficient neural network models for mobile devices and edge computing. Representative methods in this category include MobileNets[18,19], EfficientNets[32,33], and ShuffleNets[20,21]. The most prominent feature of MobileNets is the use of depthwise separable convolutions instead of ordinary convolutions to achieve model lightweighting. The characteristic of EfficientNets is that they use a compound scaling strategy to design the network, controlling the model complexity by adjusting the model depth, the network width, and the image resolution. ShuffleNets follow the design concept of sparse connectivity and reduce computation and parameters by using group pointwise convolution and channel shuffle. In addition, GhostNet[34] proposed by Huawei is also an excellent lightweight network, but the design ideas and technology used in the model are similar to those mentioned above and will not be elaborated on here.

The research on lightweight models for SOD is still in its infancy. Currently, the more representative ones include SAMNet and CSNet proposed by Professor Cheng *et al.* at Nankai University, and ELWNet[35] proposed by Professor Zhang *et al.* at Northeastern University. Among them, ELWNet is achieved through feature domain conversion, CSNet realizes model lightweighting based on the dynamic weight decay pruning method, and SAMNet is achieved by optimizing the network structure.

Currently, despite numerous studies on model lightweighting, there are still three problems: (1) Pruning, quantization, and knowledge distillation greatly influence model performance, making the model performance insufficient to meet actual needs; (2) The lightweight backbone network has a small feature domain and cannot cope with complex detection scenarios; (3) Research on lightweight models for SOD is still in its infancy. To address these issues, we propose a more efficient and lightweight SOD model - SANet.

2.3. CapsNet

CNN has a dominant position in solving computer vision-related problems. However, it discards much valuable information in the pooling process, such as the pose and position of the target. Another disadvantage of

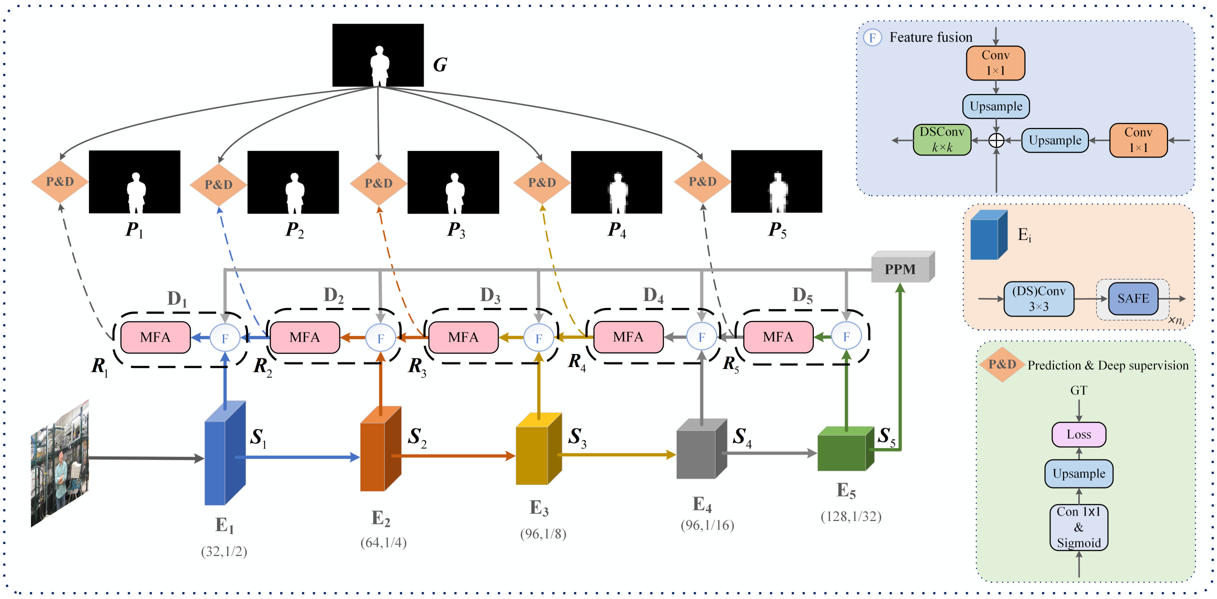

Figure 3. Overall encoder-decoder architecture of the proposed SANet. E_i represents the encoder at the i-th stage. D_i indicates the decoder at the i-th stage. S_i and R_i denote the output feature maps of the encoder and the decoder at the i-th stage, respectively. P_i stands for the predicted saliency map, and P_1 is the final prediction result. G is the ground-truth saliency map. PPM: Pyramid pooling module; MFA: multi-scale feature aggregation.

CNN is that it lacks rotation invariance and therefore requires a large amount of training data[36]. To this end, Hinton of Google Brain proposed the capsule network (CapsNet) that can capture structural information[37]. CapsNets designed a clever dynamic routing algorithm to capture the part-whole relationship in the image to enhance the equivalence of the network. Based on this advantage, related research and applications based on the CapsNet structure have been proposed one after another. Saqur *et al.* proposed a new algorithm CapsGAN by showing the weakness of the CNN-based generative adversarial network (GAN) architecture in generating 3D images[38]. Cheng *et al*. proposed complex-valued dense CapsNet (Cv-CapsNet) and complex-valued diverse CapsNet (Cv-CapsNet++) for image classification[39]. Sun *et al*. proposed a deep tensor capsule network that uses a new tensor capsule-based routing algorithm and the corresponding convolution operation[40].

Due to the unique advantages of CapsNet, it has also been successfully applied to the task of SOD. For example, Liu *et al*. optimized deep unsupervised SOD by using the part-whole relationship characteristics of CapsNet[41]. Zhang *et al*. used the attention mechanism to interact with CNN and CapsNet features to better detect salient objects[42]. Liu *et al*. integrated the advantages of CNN and CapsNet, extracted different semantic information respectively, and interacted with each other to generate better saliency prediction maps[43]. The design of the SAFE module in this paper also contains some ideas for CapsNet.

3. THE PROPOSED METHOD

In this section, the proposed SOD framework is presented. Section 3.1 describes the overall network structure. Section 3.2 introduces the SAFE module, which can adaptively extract and filter features according to the scale differences of salient objects. Section 3.3 explains the decoder design based on the MFA module.

3.1. Overall network architecture

As shown in Figure 3, the overall network structure of SANet comprises a bottom-up encoder, a top-down decoder, and a lateral connection between them. The encoder is built with SAFE modules as units and is divided into five stages. In these five stages, we downsample the input using dilated DSConv3×3 with a stride

Table 1. Backbone settings of the proposed SANet

Stage	Resolution	Module	#C	Stride	#P
E_1	336×336	DSConv3×3	32	2	-
		SAFE×1	32	1	3 (1,2,4)
E_2	168×168	DSConv3×3	64	2	-
		SAFE×1	64	1	3 (1,2,4)
E_3	84×84	DSConv3×3	96	2	-
		SAFE×3	96	1	3 (1,2,4)
E_4	42×42	DSConv3×3	96	2	-
		SAFE×6	96	1	3 (1,2,4)
E_5	21×21	DSConv3×3	128	2	-
		SAFE×3	128	1	2 (1,2)

"#C" represents the number of channels. "#P" indi-cates the number of branches of the SAFE module at each stage and the corresponding dilation rate. SAFE: Scale-adaptive feature extraction.

of 2 and adjust the number of channels. Then, we use the proposed SAFE module for scale-adaptive learning. Since the resolution of the feature map is high in the first two stages (E_1 and E_2), only a single SAFE module is used to process the feature map to reduce the computational burden. In the third to fifth stages (E_3, E_4, and E_5), we stack multiple SAFE modules to increase the receptive field and enhance the deep network representation capability. The default parameter settings of the SANet backbone network are shown in Table 1. We pass the output of the last encoder stage (E_5) through a pyramid pooling module (PPM)[44] to further improve the network's learning of global features. Different from the classic encoder-decoder network structure, this paper inputs the output features of PPM into the decoders of each stage for feature fusion, so as to make full use of the semantic information in the deep layer of the network.

3.2. SAFE module

The multi-scale information of images is important for SOD, and salient objects in natural scenes are scale-variable. To adaptively extract information from different scales of images and accurately characterize salient objects, we propose the SAFE module, which is mainly divided into two parts: multi-scale feature interaction and dynamic selection.

Multi-scale Feature Interaction: In this part, as shown in Figure 4, we first use multiple depthwise separable convolutions with different dilation rates to process the input feature map and divide the input features into various branches. Since each branch has distinct sensitivities to information of different scales, to improve the representation capabilities of different branches, we perform cross-scale feature interaction.

Specifically, let $X \in \mathbb{R}^{C\times H\times W}$ be the input feature map whose number of channels, height, and width are C, H, and W, respectively. So, we will get the feature map X_i of each branch, namely,

$$X_i = \mathcal{S}_i(X), i = 1, 2, ..., N, \tag{1}$$

where $\mathcal{S}_i$ denotes depthwise separable conv3×3 (DSConv3×3 for short) with different dilation rates at branch i, and N is the number of branches. Next, except for X_N, each feature map X_{i-1} is first processed by the 3×3 average pooling operation, and then added to X_i to obtain X_i', so X_i' can be expressed as follows:

$$X_i' = \begin{cases} X_i + AP\left(X_{i-1}\right), i = 2 \\ X_i + AP\left(X_{i-1}'\right), i = 3, ..., N, \end{cases} \tag{2}$$

where AP denotes 3×3 average pooling operation. In this way, each feature map X_i' can receive the feature information of all its previous feature maps $\left\{X_j, j \leqslant i\right\}$, which realizes feature embedding and improves the representation ability of the intra-layer branches.

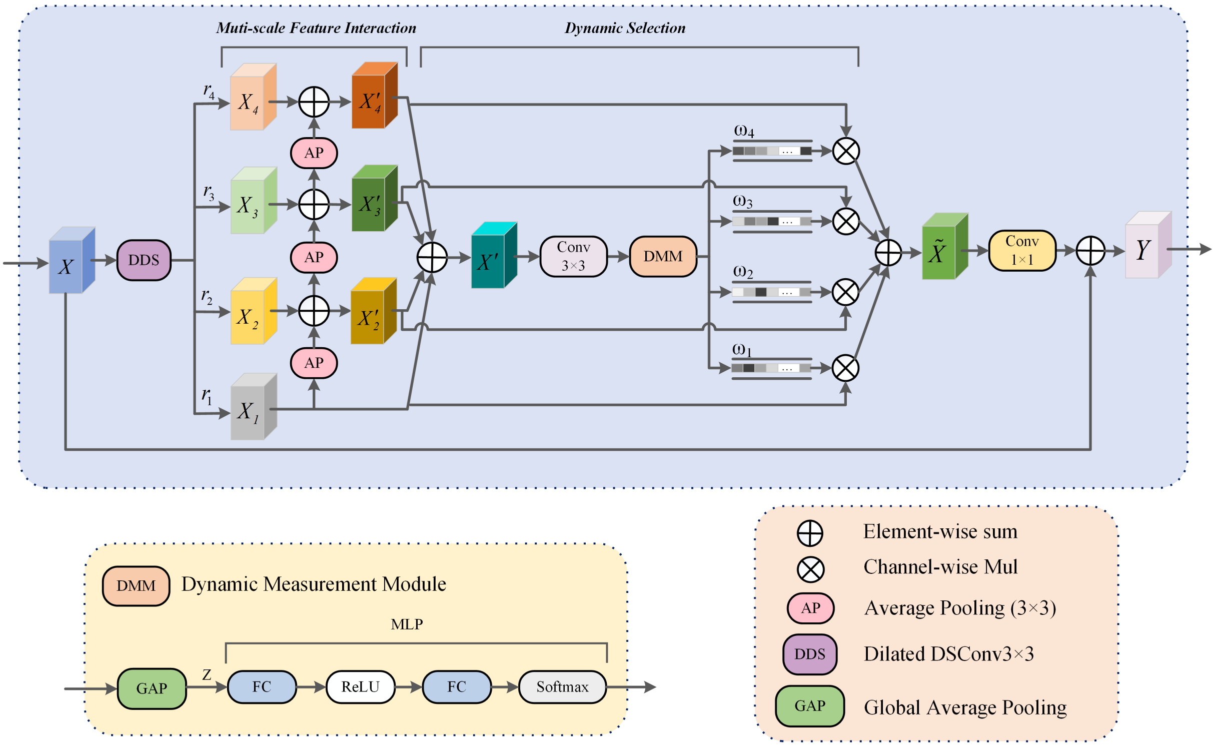

Figure 4. Illustration of the proposed SAFE module. SAFE: Scale-adaptive feature extraction.

Dynamic selection: Features of different scales have varying representation capabilities for salient objects. To measure this difference, we perform dynamic selection after completing the multi-scale feature interaction. We use an element-wise summation to integrate the feature maps output by different branches, namely,

$$X' = X_1 + \sum_{i=2}^{N} X'_i. \tag{3}$$

Here we use element-wise summation instead of concatenation because the concatenation operation will greatly increase the number of channels, resulting in heavier computational complexity and more network parameters. Next, we process X' with a 3×3 convolution and then perform the dynamic measurement module (DMM) operation. DMM consists of a global average pooling (GAP) operation and an MLP. We gather the global contextual information with channelwise statistics by using GAP. This process embeds the input X' to a learnable latent vector $Z \in \mathbb{R}^{C\times1\times1}$ by performing GAP on X' over the spatial dimension. Thus, the c-th component of Z can be given as follows:

$$Z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} X'(c, i, j), c = 0, 1, ..., C - 1. \tag{4}$$

where H stands for height, which represents the number of pixels of X' in the vertical direction, and W stands for width, which represents the number of pixels of X' in the horizontal direction. Because each element in Z indicates the importance of the corresponding feature slice of X', Z can be used as the channel-wise attention of all branches. Next, we shall perform an additional embedding regarding Z via a MLP consisting of two fully-connected layers, a non-linearity ReLU, and a softmax operation. After the MLP, a vector of size $(N \times C) \times 1 \times 1$ will be output, and then we will split it into N parts corresponding to N different branches through the split operation, and the i-th part is $\omega_i \in \mathbb{R}^{C\times1\times1}$. Since MLP is learnable, different attention weights can be dynamically assigned to each scale feature. The dynamic attention weight ω_i of the i-th branch

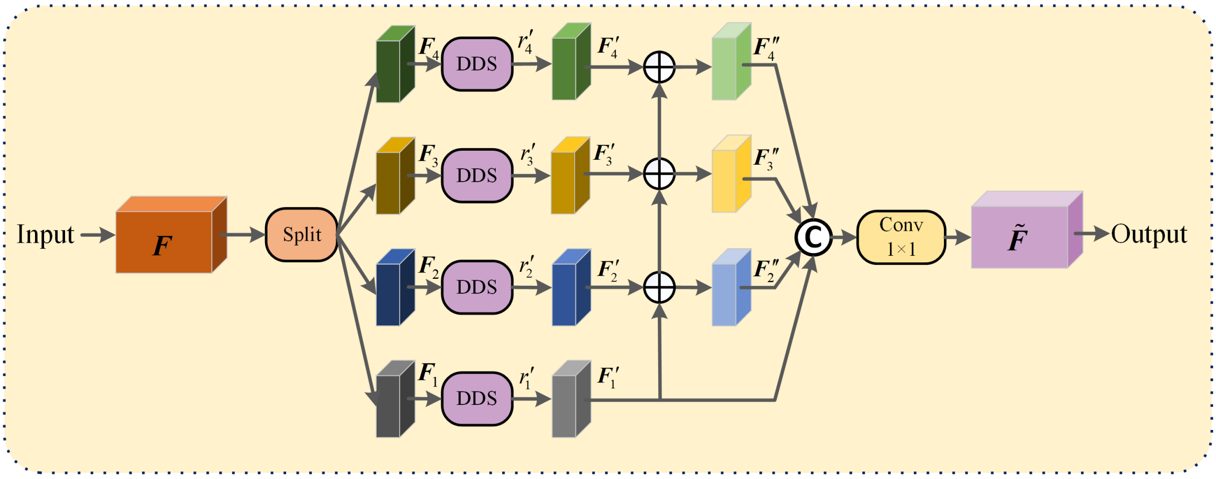

Figure 5. Illustration of the proposed MFA module. MFA: Multi-scale feature aggregation; DDS: dilated DSConv3×3 operation; Ⓒ: a concatenation operation.

is calculated as follows:

$$\omega_i = \text{Split}\left(Softmax\left(FC\left(ReLU\left(FC\left(Z\right)\right)\right)\right)\right), i = 1, 2, ..., N \tag{5}$$

where $\omega_i \in \mathbb{R}^{C\times 1\times 1}$. After that, we use channel-wise multiplication to combine ω_i with the corresponding branch features and integrate the multiplied results by element-wise summation to obtain the feature map $\tilde{X}$:

$$\tilde{X} = \omega_1 \times X_1 + \sum_{i=2}^{N} \left(\omega_i \times X_i'\right). \tag{6}$$

After $\tilde{X}$ undergoes a Conv1×1, it is added to the original input feature map X through a residual connection to obtain the final output Y, namely,

$$Y = \kappa\left(\tilde{X}\right) + X, \tag{7}$$

where κ denotes a Conv1×1 operation.

As shown in Figure 3, the SAFE module is the basic unit that forms the backbone network of SANet. The number of branches N mentioned above is set as a hyperparameter. In the subsequent ablation experiments, we prove that the higher the resolution of the input feature map, the more branches are needed, so we will set different numbers of branches at multiple stages of the network.

3.3. MFA-based decoder

This paper first uses the backbone network composed of SAFE modules to extract image features, and then we design the decoder. As shown in Figure 3, the decoder consists of two parts: feature fusion and MFA modules.

The feature fusion module fuses features from three directions and uses a depthwise separable $k\times k$ dilated convolution to integrate the fused results. After feature fusion, it is not enough to simply use a layer of convolution for processing. In particular, the output features of the PPM module will go through a relatively large scale span (maximum span of 16 times) during the upsampling process, and it is necessary to establish connections across scales reasonably. Therefore, we designed a MFA module to process further the fused features.

As shown in Figure 5, we use the output feature map of the feature fusion module as the input of the MFA module and split it into four parts by channel, represented by F_i, where $i \in \{1, 2, 3, 4\}$. Then, information on different scales is extracted through a depth-separable convolution with a dilation rate of r_i', forming four different branches and obtaining feature maps F_i', where $r_i' \in \{1, 2, 3, 4\}$. So, F_i' can be expressed as follows:

$$F_i' = \$_i \left(F_i\right), i = 1, 2, 3, 4, \tag{8}$$

where $\$_i$ represents DSConv3×3 with different dilation rates at branch i. To achieve cross-scale feature aggregation, we use residual connections between different branches and get F_i'' by element-wise summation:

$$F_i'' = \begin{cases} F_i' + F_{i-1}', i = 2 \\ F_i' + F_{i-1}'', i = 3, 4. \end{cases} \tag{9}$$

Then, each branch's results are merged as output through a concatenation operation, namely,

$$\tilde{F} = \kappa \left(Concat \left(F_1', F_2'', F_3'', F_4''\right)\right). \tag{10}$$

Let R_i represent the feature maps output by the decoder at each stage, and S_i represent the feature maps output by the encoder at each stage, where $i \in \{1, 2, 3, 4, 5\}$. So,

$$R_5 = \mathrm{MFA}\left(\zeta_5^{k\times k}\left(Up\left(\kappa\left(\mathrm{PPM}\left(S_5\right)\right)\right) + S_5\right)\right), \tag{11}$$

where MFA represents MFA module, $\zeta_5^{k\times k}$ means DSConv$k\times k$ at the fifth stage, and Up indicates upsampling operation. In summary, we have

$$R_i = \mathrm{MFA}\left(\zeta_i^{k\times k}\left(Up\left(\kappa\left(\mathrm{PPM}\left(S_5\right)\right)\right) + Up\left(\kappa\left(R_{i+1}\right)\right) + S_i\right)\right), i = 1, 2, 3, 4. \tag{12}$$

where $\zeta_i^{k\times k}$ represents DSConv$k\times k$ at the i-th stage.

3.4. Saliency reasoning

We use deep supervision to improve the transparency of the hidden layer learning process. As shown in Figure 3, for the fusion features at different stages, we use a Conv1×1 and sigmoid activation function to generate multiple predictions, namely P_i, where $i \in \{1, 2, 3, 4, 5\}$. We adopt the standard binary cross-entropy loss for training, which is defined as follows:

$$L_{Total} = L_{BCE}\left(P_1, G\right) + \lambda \sum_{i=2}^{5} L_{BCE}\left(P_i, G\right), \tag{13}$$

where L_{BCE} is the standard binary cross-entropy loss function, and G denotes the ground-truth saliency map. λ denotes the weighting scalar for loss balance, which is set to 0.4.

4. RESULTS

4.1. Experimental setup

4.1.1 Implementation details

This paper uses the PyTorch library to implement the proposed method. Our model is pre-trained on the ImageNet dataset. The training set of the DUTS[45] dataset (DUTS-TR) is used for model training. In addition, we also validate our proposed method on the traffic dataset TSOD, using its first 2,000 images for training and the rest for testing. All experiments are performed using the Adam optimizer, with parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay of 10^{-4}, and batch size of 20. We use poly learning rate scheduler so that the learning rate for the n-th epoch is $init\_lr \times \left(1 - \frac{n}{\#epochs}\right)^{power}$, where $init\_lr = 5 \times 10^{-4}$ and $power = 0.9$. We trained the proposed model for 300 epochs, i.e., $\#epochs =$ 300. All experiments are run on a server with an NVIDIA GTX3090 GPU and an AMD Ryzen Threadripper 3960X (2.2GHz) CPU.

4.1.2 Datasets

We validate our proposed method on six common datasets, including DUTS, DUT-OMRON[46], ECSSD[47], PASCAL-S[48], HKU-IS[49], and SOD[50]. In addition, we also verified the advantages of our method over other SOD methods in the traffic field in the traffic dataset TSOD.

The DUTS dataset comprises two subsets: the training set, DUTS-TR, and the test set, DUTS-TE. DUTS-TR is used for SANet training, while DUTS-TE is reserved for testing. DUTS-TR includes 10,553 images from ImageNet, each annotated at the pixel level. The test set contains 5,019 images selected from ImageNet and SUN and their pixel-level labels. DUT-OMRON features 5,168 images depicting complex scenes with rich contents, accompanied by pixel-level labels. ECSSD consists of 1,000 images, with pixel-level labels, presenting a high level of interference in both the foreground and background of the images, making it a challenging dataset. PASCAL-S includes 850 images and their pixel-level labels, showcasing relatively complex scenes. HKU-IS contains 4,447 images and their pixel-level labels, and almost all images have multiple salient objects. SOD contains 300 images and their pixel-level labels, where the color contrast between salient objects and the background is low. TSOD consists of 2,316 images of traffic scenes with relatively complex content, along with their pixel-level labels.

4.1.3 Evaluation criteria

This paper evaluates the effectiveness of the proposed model using the maximum F-measure (maxF), average F-measure (avgF), mean absolute error (MAE), and S-measure (S). Additionally, the efficiency of the model is assessed through the number of model parameters (#Param), the number of FLOPs, and the FPS.

F-measure is an evaluation method that comprehensively considers precision and recall, which is defined as follows:

$$F_\beta = \frac{\left(1+\beta^2\right) \times P \times R}{\beta^2 \times P + R}, \tag{14}$$

where P and R represent precision and recall, respectively. We set $\beta^2 = 0.3$ to emphasize the importance of precision.

MAE aims to measure the difference between the predicted image P and the ground truth G, which is calculated as follows:

$$\mathrm{MAE}\left(P, G\right) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \left|P_{ij} - G_{ij}\right|, \tag{15}$$

where H and W represent the height and width of the saliency map, respectively, and P_{ij} and G_{ij} represent the pixel values of the i-th row and j-th column of P and G.

S is used to evaluate the structural similarity between the predicted saliency map and the ground truth and is calculated by:

$$S = \alpha \times S_0 + (1-\alpha) \times S_r, \tag{16}$$

where S_0 represents the target structure similarity, S_r represents the regional structure similarity, and α is set to 0.5.

In this paper, #Param is measured in million (M) and FLOPs is measured in giga (G). FLOPs are used to measure the computational effort of the model. FPS indicates the number of images that the model can infer per second when using an NVIDIA GTX3090 GPU. For all SOD methods, we use 336×336 input and the same hardware and training strategy.

Table 2. Comparison with existing methods in terms of #Param, FLOPs, FPS, maxF, avgF, MAE, and S in general scenarios

Methods	#Param	FLOPs	FPS	DUTS-TE				DUT-OMRON				ECSSD			
	(M)	(G)		maxF↑	avgF↑	MAE↓	S↑	maxF↑	avgF↑	MAE↓	S↑	maxF↑	avgF↑	MAE↓	S↑
Heavyweight method (#Param > 10 M)															
CPD [51]	47.85	59.5	42	0.861	0.805	0.043	0.866	0.794	0.747	0.056	0.818	0.930	0.917	0.037	0.905
U2Net [52]	44.02	58.8	45	0.873	0.792	0.045	0.874	0.823	0.761	0.055	0.847	0.951	0.892	0.033	0.928
UCF [53]	29.47	61.4	12	0.772	0.631	0.112	0.782	0.730	0.621	0.120	0.760	0.901	0.844	0.069	0.883
Amulet [25]	33.15	45.3	10	0.778	0.678	0.085	0.804	0.743	0.647	0.098	0.781	0.913	0.868	0.059	0.894
DSS [54]	62.23	114.6	7	0.825	0.720	0.056	0.826	0.781	0.656	0.066	0.790	0.921	0.842	0.056	0.879
PiCANet [13]	32.85	19.7	5	0.851	0.759	0.051	0.869	0.794	0.717	0.065	0.832	0.931	0.886	0.046	0.917
BASNet [55]	87.06	127.3	36	0.859	0.791	0.048	0.866	0.805	0.756	0.057	0.836	0.938	0.880	0.037	0.916
PoolNet [56]	53.63	123.4	39	0.866	0.819	0.043	0.875	0.791	0.752	0.057	0.829	0.934	0.919	0.048	0.909
MINet [14]	162.38	87.1	43	0.877	0.823	0.039	0.875	0.794	0.741	0.057	0.822	0.943	0.922	0.036	0.919
VST [57]	44.48	23.2	40	0.877	0.818	0.037	0.896	0.800	0.756	0.058	0.850	0.944	0.920	0.033	0.932
PFSNet [58]	31.18	37.6	44	0.898	0.846	0.036	0.890	0.823	0.774	0.055	0.852	0.952	0.932	0.031	0.927
ICON [59]	33.09	20.9	57	0.892	0.838	0.037	0.888	0.825	0.772	0.057	0.844	0.950	0.928	0.032	0.929
MENet [60]	-	-	45	0.912	0.893	0.028	0.905	0.834	0.818	0.045	0.850	0.955	0.942	0.031	0.928
TSERNet [61]	189.64	203.6	35	0.861	0.798	0.046	0.864	0.818	0.768	0.056	0.837	0.945	0.922	0.031	0.930
A3Net [62]	17.00	34.1	46	0.843	0.769	0.052	0.863	0.801	0.739	0.062	0.831	0.937	0.913	0.045	0.912
Avg-heavy	62.00	72.6	34	0.856	0.785	0.051	0.863	0.797	0.735	0.064	0.825	0.936	0.902	0.042	0.914
Lightweight method (#Param <= 10 M)															
HVPNet [16]	1.24	1.1	55	0.839	0.749	0.058	0.849	0.799	0.721	0.065	0.831	0.925	0.889	0.052	0.904
SAMNet [17]	1.33	**0.5**	37	0.835	0.745	0.058	0.849	0.797	0.717	0.065	0.830	0.925	0.891	0.050	0.907
CSNet [15]	**0.14**	1.5	48	0.819	0.687	0.074	-	0.792	0.675	0.081	-	0.916	0.844	0.065	-
Ours	2.29	1.5	**62**	**0.845**	**0.773**	**0.054**	**0.866**	**0.804**	**0.742**	**0.061**	**0.833**	**0.934**	**0.907**	**0.047**	**0.913**

Methods	#Param	FLOPs	FPS	DUTS-TE				DUT-OMRON				ECSSD			
	(M)	(G)		maxF↑	avgF↑	MAE↓	S↑	maxF↑	avgF↑	MAE↓	S↑	maxF↑	avgF↑	MAE↓	S↑
Heavyweight method (#Param > 10 M)															
CPD [51]	47.85	59.5	42	0.866	0.820	0.071	0.847	0.924	0.891	0.034	0.904	0.848	0.740	0.113	0.765
U2Net [52]	44.02	58.8	45	0.859	0.770	0.074	0.845	0.935	0.896	0.031	0.916	0.861	0.769	0.106	0.789
UCF [53]	29.47	61.4	12	0.757	0.726	0.116	0.806	0.888	0.823	0.062	0.875	0.805	0.737	0.148	0.763
Amulet [25]	33.15	45.3	10	0.806	0.757	0.100	0.818	0.897	0.841	0.051	0.886	0.795	0.741	0.144	0.755
DSS [54]	62.23	114.6	7	0.831	0.740	0.101	0.820	0.916	0.844	0.041	0.881	0.846	0.747	0.122	0.746
PiCANet [13]	32.85	19.7	5	0.880	0.792	0.076	0.854	0.921	0.870	0.043	0.904	0.855	0.785	0.103	0.793
BASNet [55]	87.06	127.3	36	0.854	0.771	0.076	0.838	0.928	0.896	0.032	0.909	0.849	0.744	0.112	0.772
PoolNet [56]	53.63	123.4	39	0.855	0.826	0.065	0.867	0.925	0.903	0.037	0.908	0.863	0.758	0.111	0.781
MINet [14]	162.38	87.1	43	0.882	0.843	0.065	0.855	0.932	0.906	0.030	0.914	-	-	-	-
VST [57]	44.48	23.2	40	0.850	0.829	0.061	0.873	0.937	0.900	0.029	0.928	0.866	0.833	0.065	0.854
PFSNet [58]	31.18	37.6	44	0.881	0.837	0.063	0.876	0.943	0.919	0.026	0.933	-	-	-	-
ICON [59]	33.09	20.9	57	0.876	0.833	0.064	0.861	0.940	0.910	0.029	0.920	0.879	0.804	0.084	0.824
MENet [60]	-	-	45	0.890	0.870	0.054	0.872	0.948	0.932	0.023	0.927	0.878	0.868	0.087	0.809
TSERNet [61]	189.64	203.6	35	0.857	0.782	0.062	0.840	0.930	0.904	0.036	0.910	0.850	0.746	0.109	0.775
A3Net [62]	17.00	34.1	46	0.844	0.791	0.089	0.831	0.920	0.881	0.042	0.903	0.843	0.787	0.120	0.765
Avg-heavy	62.00	72.6	34	0.853	0.799	0.076	0.847	0.926	0.888	0.036	0.908	0.849	0.774	0.110	0.784
Lightweight method (#Param <= 10 M)															
HVPNet [16]	1.24	1.1	55	0.826	0.784	0.089	0.830	0.915	0.872	0.045	0.899	0.826	0.779	0.122	0.765
SAMNet [17]	1.33	**0.5**	37	0.812	0.778	0.092	0.826	0.915	0.871	0.045	0.898	0.833	0.780	0.124	0.762
CSNet [15]	**0.14**	1.5	48	0.835	0.723	0.103	-	0.899	0.840	0.059	-	0.825	0.724	0.137	-
Ours	2.29	1.5	**62**	**0.847**	**0.801**	**0.084**	**0.833**	**0.919**	**0.889**	**0.044**	**0.901**	**0.845**	**0.796**	**0.117**	**0.767**

The larger the mF and S, the better, the smaller the MAE, the better, and Avg-heavy represents the average of each metric of all heavyweight methods. The best lightweight methods are in bold, and the underline indicates the metrics where SANet is better than Avg-heavy. FLOPs: Floating-point operations; FPS: frames per second; MAE: mean absolute error.

4.2. Performance analysis

In this section, we compare SANet with eighteen typical SOD methods, including fifteen heavyweight methods and three lightweight state-of-the-art methods. This paper uses the same method to evaluate the detection results of related models.

4.2.1 Comparison with heavyweight SOD methods in general scenarios

Table 2 shows the evaluation results of SANet and existing state-of-the-art SOD methods in terms of #Param, FLOPs, FPS, maxF, avgF, MAE, and S. From Table 2, we can see that SANet can achieve the performance of general heavyweight methods in the four evaluation metrics of maxF, avgF, MAE, and S. Especially in the challenging DUT-OMRON dataset, the four performance metrics all exceed the average level of heavyweight methods, with maxF, avgF, and S increase by 0.88%, 0.95%, and 0.97%, respectively, and MAE reduced by 4.92%. In terms of efficiency metrics, SANet has reduced parameters by 96.31%, reduced FLOPs by 97.93%, and increased FPS by 82.35% compared to the average level of heavyweight methods.

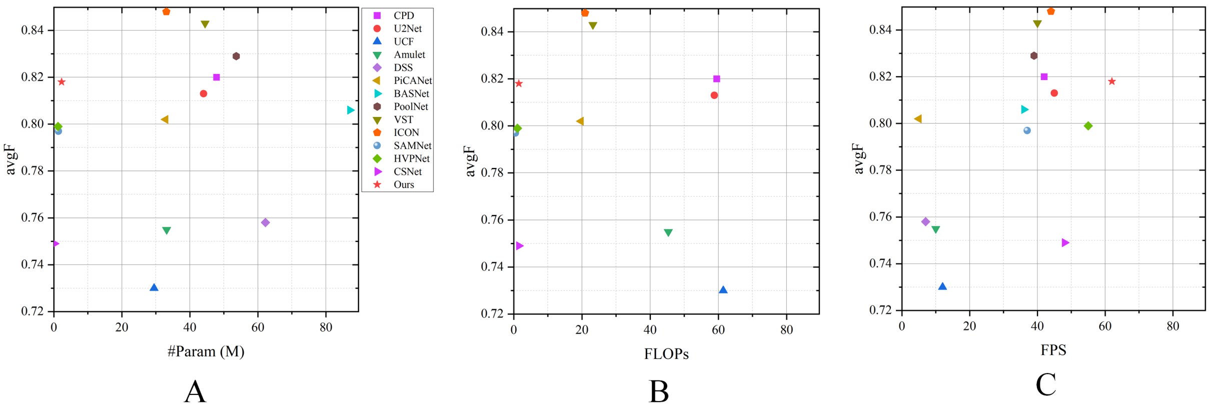

Figure 6. Illustration of the trade-off between performance and efficiency. The avgF is the average of the results on the six datasets. (A) avgF *vs*. #Param; (B) avgF *vs*. FLOPs; (C) avgF *vs*. FPS. FLOPs: Floating-point operations; FPS: frames per second.

Compared with integrity cognition network (ICON), SANet reduces parameters and FLOPs by 93.08% and 92.82%, respectively, and increases the FPS by 8.77%, while the average values of maxF and avgF on the six datasets only decrease by 3.13% and 3.54%, respectively.

4.2.2 Comparison with lightweight SOD methods in general scenarios

Table 2 also shows the quantitative comparison results of SANet and other state-of-the-art lightweight SOD models, including HVPNet, SAMnet, and CSNet. Compared with SAMNet, the maxF of the proposed model on the six datasets is improved by 1.20%, 0.88%, 0.97%, 4.31%, 0.44%, and 1.44%, respectively, and the avgF is improved by 3.76%, 3.49%, 1.80%, 2.96%, 2.22%, and 2.05%, respectively, and the FPS is improved by 67.57%. As we can see, although SANet does not reach the optimal level regarding #param and FLOPs, SANet far exceeds the above lightweight models in terms of maxF, avgF, MAE, and S. It should be emphasized that SANet achieves a FPS far exceeding that of other models.

4.2.3 Comprehensive comparison in general scenarios

Figure 6 shows the comprehensive comparison results of this paper and other methods in terms of model performance and efficiency. In the sub-figures of Figure 6A and B, SANet lies at the top-left corner. In Figure 6C, it lies at the top-right corner. This shows that SANet achieves higher accuracy with fewer parameters and FLOPs and faster speed. Therefore, it achieves a good trade-off between performance and efficiency.

4.2.4 Qualitative comparison in general scenarios

For practical application scenarios of SOD, good visual qualitative effects are sometimes more important than quantitative performance. In Figure 7, we provide visual SOD results in five typical scenarios to evaluate the model effect. It can be seen that in the simple scene [Figure 7A], the visual detection results of SANet are comparable to those of heavyweight methods, and the depiction of salient target details is more accurate than other lightweight models. In the small target scene [Figure 7B], heavyweight methods, boundary-aware SOD network (BASNet), PoolNet, and visual saliency transformer (VST), can accurately identify salient targets, while cascaded partial decoder (CPD), SOD using short connections (DSS), ICON, and MENet have false positives and false negatives. Our model can segment small targets, and the boundaries are clearer than other lightweight methods, without false positives and false negatives. This is also due to the SAFE module, which enables our model to adaptively capture salient objects of any size. In the low-contrast scene [Figure 7C], DSS and PoolNet have false positives, while other heavyweight methods can accurately identify salient objects. Among lightweight methods, CSNet has false positives, whereas SANet, SAMNet, and HVPNet do not. However, compared with SANet, SAMNet and HVPNet do not accurately depict the details of salient objects. In

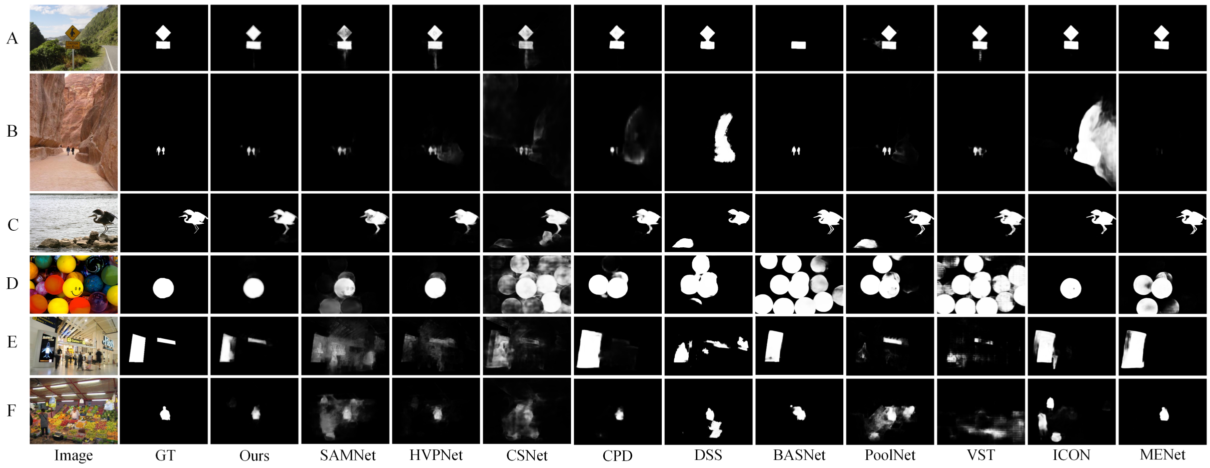

Figure 7. Visual comparison with mainstream SOD methods in general scenarios. (A) Simple scene; (B) small target scene; (C) low contrast scene; (D) confusing scene; (E) complex scene; (F) complex scene. SOD: Salient object detection.

Table 3. The Comparison of maxF, avgF, MAE, and S in traffic scenarios

Methods	TSOD			
	maxF↑	avgF↑	MAE↓	S↑
SAMNet	0.333	0.126	0.058	0.590
CSNet	0.233	0.055	0.062	0.535
Ours	0.650	0.347	0.036	0.700

MAE: Mean absolute error; TSOD: traffic salient object detection.

the confusing scene [Figure 7D], SANet can still accurately identify the target object, while other methods except ICON have false positives. In the complex scene [Figure 7E and F], other lightweight methods including some heavyweight methods have false positives and negatives, while our method has demonstrated excellent performance in the complex scene regardless of whether there is one or multiple salient targets.

4.2.5 Comparison in traffic scenarios

We use three models in TSOD for comparative experiments, and they perform well in general scenarios. These models are trained on the TSOD dataset; the final test results are shown in Table 3 and Figure 8. From Table 3, we can see that our method is better than the comparison methods in terms of maxF, avgF, MAE, and S. As shown in Figure 8, we qualitatively compare our method with two other excellent lightweight SOD methods in seven different scenarios. Figure 8A is a simple scene. It can be seen that SANet can easily identify the salient target and depict its outline clearly. In contrast, the performance of the other two methods is unsatisfactory. Figure 8B and C are small target scenes during the day. It can be seen that SANet still performs well for such ultra-small targets, thanks to the fact that our proposed SAFE module is sensitive to objects of various scales. Figure 8D is a multi-target scene. It can be seen that SANet does not mistakenly regard the large truck next to it as a salient target, but accurately identifies the relatively small correct salient target in the distance. Figure 8E is a multi-target scene with a complex background. SANet can also identify the unique salient target, while SAMNet does not identify any salient objects, and CSNet includes other targets. Figure 8F is a night scene with low contrast and interference from vehicle lights. SANet can accurately identify road signs, while the other two methods do not identify any salient objects. Figure 8G is a small target scene at night. Although SANet did not accurately identify the outline of the small target, it still correctly identified the road sign. In general, compared with the other two excellent lightweight SOD methods, SANet can obtain better results in general

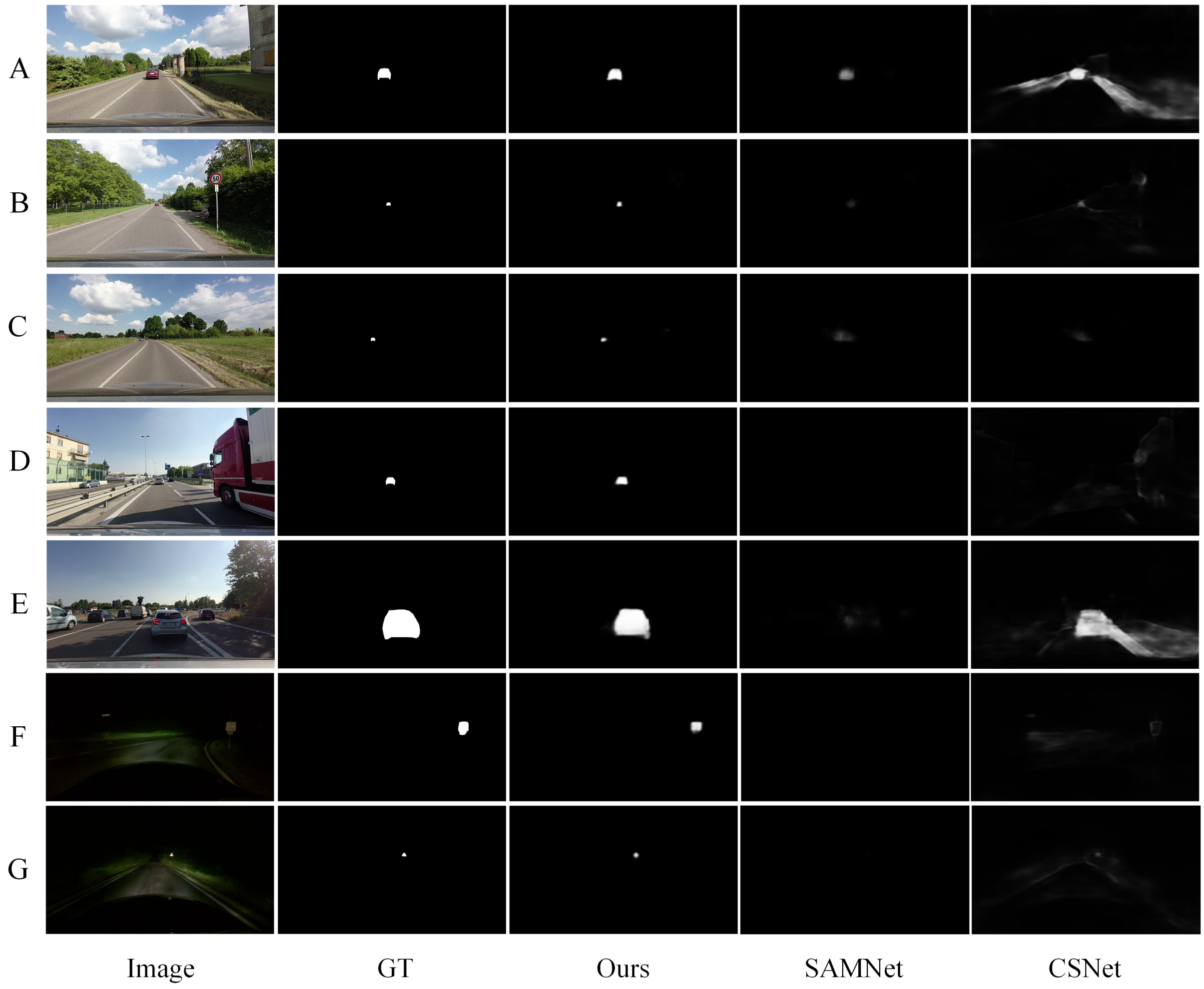

Figure 8. Visual comparison of different methods in traffic scenarios. (A) Simple scene during daytime; (B) small target scene during daytime; (C) small target scene during daytime; (D) multi-target scene; (E) multi-target and complex background scene; (F) low contrast scene at night; (G) small target scene at night.

scenes and various challenging scenes, which also proves the effectiveness of our proposed innovations.

4.3. Failure cases

Although our proposed method achieves excellent performance in multiple scenarios, it does encounter some failure cases. As shown in Figure 9, we provide failure cases in different scenarios and conduct an in-depth analysis. From Figure 9A, we can see that SANet did not identify any significant targets, showing that SANet's recognition ability for ultra-small targets is still limited. In Figure 9B, due to the low overall contrast between the vehicle and the background, SANet did not identify the vehicle as a significant target, but mistakenly identified the roadside sign. Figure 9C and D are multi-target scenes. SANet both mistakenly identified multiple targets and the sizes of the acquired targets were somewhat different. This is because SANet has a strong ability to distinguish objects of different scales. We will improve it in future experiments. Figure 9E is a multi-target scene with a complex background. It can be seen that SANet still mistakenly identified multiple targets, and the recognition accuracy is not high when two vehicles overlap. Figure 9F is a small target scene at night. It can be seen that the headlights will have a certain impact on the detection results. Figure 9G is a scene with strong light interference at night. In this scene, the noise interference is strong, which has a certain impact on the detection results. We attribute these failure cases to the following factors: (1) The number of images in the training set is limited and cannot cover all traffic scenarios; (2) The actual traffic scenarios are complex

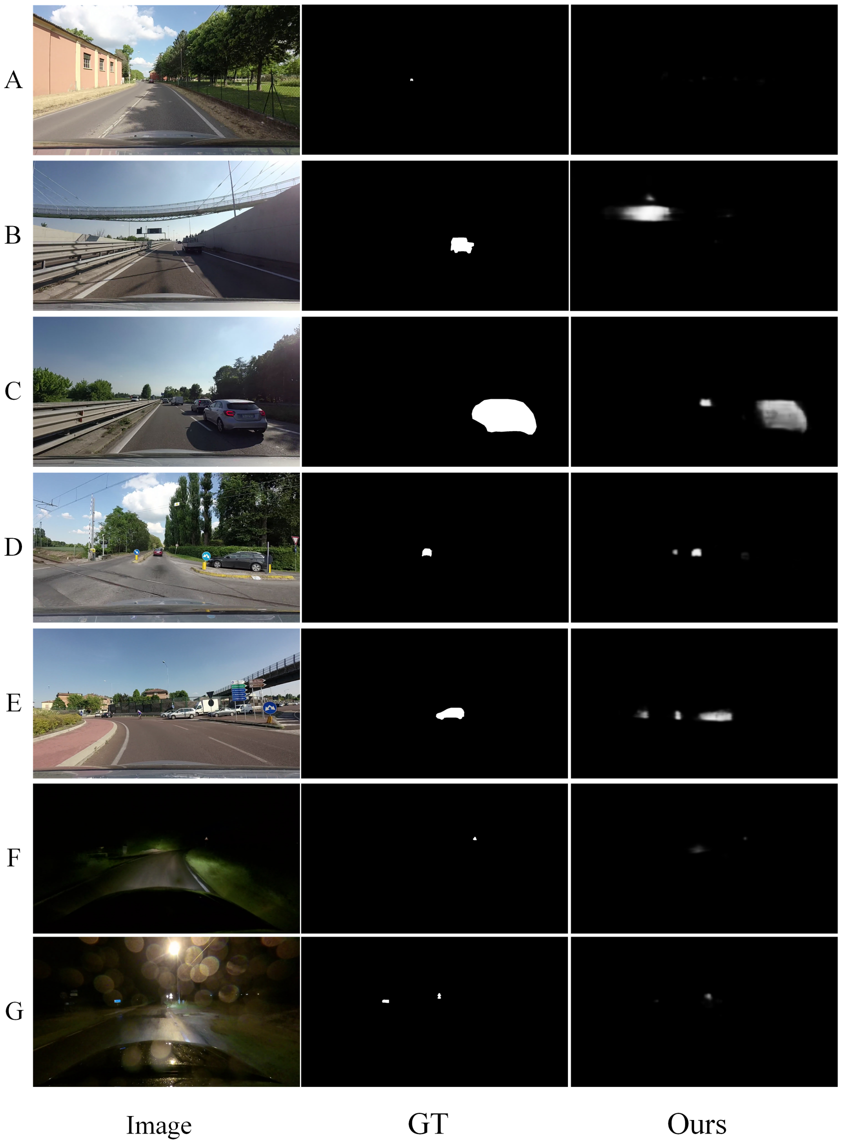

Figure 9. Failure cases of SANet. (A) Small target scene during daytime; (B) low contrast scene during daytime; (C) multi-target scene; (D) multi-target scene; (E) multi-target and complex background scene; (F) small target scene at night; (G) strong light interference scene at night.

and diverse, which poses a huge challenge to the performance of the model; (3) The representation ability of lightweight networks is limited, and the network's processing capabilities for overly complex traffic scenarios are still insufficient.

The model we proposed currently performs poorly in traffic detection, especially in small object scenes and complex background scenes. To address this issue, we propose several possible solutions in the future: (1) The training set of the traffic scene SOD dataset TSOD used in this study has only 2,000 images, which is less than one-fifth of DUTS-TR. In the future, we can improve it by increasing the number of training set images; (2) Currently, computing power is developing rapidly, and the computing power of onboard computing is far superior to that of the past. We may be able to improve the model's representation capabilities for small target scenes and complex scenes by appropriately increasing the number of model parameters and model depth; (3) Use knowledge distillation to use large-scale networks or pre-trained models as teacher models to guide the

Table 4. Ablation study on the proposed SANet components

Ver.	Methods	DUTS-TE		DUT-OMRON		ECSSD		PASCAL-S		HKU-IS		SOD	
		maxF↑	MAE↓	maxF↑	MAE↓	maxF↑	MAE↓	maxF↑	MAE↓	maxF↑	MAE↓	maxF↑	MAE↓
0	Basic	0.819	0.068	0.779	0.076	0.911	0.069	0.825	0.112	0.898	0.056	0.821	0.133
1	Basic+MI	0.828	0.063	0.790	0.067	0.920	0.060	0.832	0.094	0.910	0.048	0.833	0.125
2	Basic+MI+DS	0.830	0.060	0.792	0.065	0.922	0.058	0.834	0.093	0.912	0.047	0.836	0.124
3	Basic+MI+DS+MF	0.834	0.059	0.794	0.064	0.924	0.055	0.836	0.088	0.913	0.048	0.842	0.121
4	Basic+MI+DS+MF+PR	0.845	0.054	0.804	0.061	0.934	0.047	0.847	0.084	0.919	0.044	0.845	0.117

We use the vanilla single branch module as the base model (Ver.0). Here, "MI", "DS", "MF", and "PR" refer to the multi-scale feature interaction, dynamic selection, MFA module, and ImageNet pre-training, respectively.

learning of lightweight networks.

4.4. Ablation study

In this section, we conduct an ablation study on the proposed module components, the backbone network's effectiveness, and the SAFE module's configuration to demonstrate our proposed model's effectiveness. The relevant experimental settings are consistent with those outlined in Section 4.1.

4.5. Proposed module components

Table 4 shows the results of the ablation study of the model components in this paper. As the number of model components increases, the model performance improves progressively. Compared with Ver.0, the average values of maxF on six datasets of Ver.3 increased by 0.015 and MAE decreased by 0.014. There is no ImageNet pre-training between Ver.0 and Ver.3, and the difference in their experimental results also shows that the proposed model is effective.

4.6. The effectiveness of the backbone network

In addition to existing SOD methods, we also compared several widely used lightweight backbone networks, including MobileNet, MobileNetV2, ShuffleNetV2, and EfficientNet. To use these lightweight backbone networks for SOD tasks, we add the same decoder as SANet to these networks for ablation study.

In Table 5, we can see that directly applying the existing lightweight backbone network to the SOD task does not produce satisfactory results regarding accuracy. Taking EfficientNet as an example, we take the average values of maxF, avgF, and MAE of six data sets. The results showed that compared to EfficientNet, SANet achieved a 13.20% improvement in maxF, an 11.14% improvement in avgF, and a 44.72% reduction in MAE. This further verifies the correctness and rationality of our redesign of the backbone network structure for SOD.

4.7. Configuration of the SAFE module

Table 6 presents the ablation study results of the SAFE module with varying branch numbers and dilation rates. Increasing the number of branches in the E_1-E_4 stages improves some metrics, but also significantly increases computational complexity, which contradicts our goal of maintaining a lightweight model. The default settings of the SAFE module are selected after weighing the trade-off between model accuracy and complexity.

5. CONCLUSION

This paper reviews existing research on SOD and analyzes the challenges in current approaches. Heavyweight SOD models face difficulties in scenarios with low computing power and high real-time requirements due to issues such as large model size and poor real-time performance. In contrast, lightweight SOD models have poor detection performance and struggle to handle complex scenarios. To address these problems, we propose SANet, a scale-adaptive lightweight SOD model that achieves a trade-off between lightweight design and detection effectiveness. We first implement the SAFE module, a component unit of the backbone net-

Table 5. Ablation study on different backbone networks

Backbone	#Param (M)	FLOPs (G)	FPS	DUTS-TE				DUT-OMRON				ECSSD			
				maxF↑	avgF↑	MAE↓	S↑	maxF↑	avgF↑	MAE↓	S↑	maxF↑	avgF↑	MAE↓	S↑
MobileNet [18]	4.27	2.2	36	0.804	0.712	0.067	0.825	0.753	0.678	0.073	0.805	0.906	0.869	0.064	0.884
MobileNetV2 [19]	2.37	0.8	47	0.798	0.708	0.066	0.823	0.758	0.675	0.075	0.806	0.905	0.865	0.066	0.885
ShuffleNetV2 [21]	**1.60**	**0.6**	33	0.743	0.698	0.071	0.816	0.720	0.666	0.076	0.797	0.870	0.861	0.069	0.878
EfficientNet [32]	8.64	2.6	44	0.723	0.687	0.112	0.748	0.696	0.656	0.105	0.778	0.848	0.826	0.104	0.783
Ours	2.29	1.5	**62**	**0.845**	**0.773**	**0.054**	**0.866**	**0.804**	**0.742**	**0.061**	**0.833**	**0.934**	**0.907**	**0.047**	**0.913**

Backbone	#Param (M)	FLOPs (G)	FPS	PASCAL-S				HKU-IS				SOD			
				maxF↑	avgF↑	MAE↓	S↑	maxF↑	avgF↑	MAE↓	S↑	maxF↑	avgF↑	MAE↓	S↑
MobileNet [18]	4.27	2.2	36	0.821	0.751	0.099	0.801	0.895	0.855	0.052	0.884	0.809	0.744	0.135	0.737
MobileNetV2 [19]	2.37	0.8	47	0.806	0.747	0.102	0.798	0.89	0.854	0.056	0.879	0.801	0.746	0.138	0.742
ShuffleNetV2 [21]	**1.60**	**0.6**	33	0.781	0.742	0.107	0.794	0.853	0.848	0.059	0.871	0.779	0.734	0.147	0.715
EfficientNet [32]	8.64	2.6	44	0.755	0.736	0.132	0.754	0.844	0.807	0.114	0.762	0.722	0.706	0.168	0.689
Ours	2.29	1.5	**62**	**0.847**	**0.801**	**0.084**	**0.833**	**0.919**	**0.889**	**0.044**	**0.901**	**0.845**	**0.796**	**0.117**	**0.767**

The best methods are in bold. FLOPs: Floating-point operations; FPS: frames per second; MAE: mean absolute error.

Table 6. Ablation study on the SAFE module configuration

Stage	#B	#D	DUTS-TE			DUT-OMRON			ECSSD			PASCAL-S			HKU-IS			SOD		
			maxF↑	avgF↑	MAE↓	maxF↑	avgF↑	MAE↓	maxF↑	avgF↑	MAE↓	maxF↑	avgF↑	MAE↓	maxF↑	avgF↑	MAE↓	maxF↑	avgF↑	MAE↓
Default configuration			0.834	0.765	0.059	**0.794**	0.734	0.064	**0.924**	**0.896**	**0.055**	0.836	0.793	0.088	0.913	0.881	0.048	0.842	0.786	0.121
E_1-E_4	2	1,2	0.830	0.762	0.061	0.788	0.730	0.068	0.920	0.893	0.057	0.831	0.787	0.091	0.908	0.874	0.052	0.837	0.756	0.130
E_1-E_4	4	1,2,3,4	0.833	0.765	0.060	0.792	0.733	0.065	0.921	0.894	0.057	0.835	0.792	0.089	0.911	0.880	0.050	0.842	**0.787**	**0.120**
E_1-E_4	4	1,2,4,8	**0.836**	**0.766**	**0.056**	0.792	**0.735**	**0.063**	0.923	0.895	0.056	**0.837**	**0.795**	**0.086**	**0.914**	**0.883**	**0.045**	**0.843**	0.786	0.121
E_5	3	1,2,4	0.831	0.762	0.062	0.787	0.728	0.069	0.918	0.892	0.060	0.832	0.789	0.088	0.907	0.872	0.055	0.834	0.753	0.132

"Default configuration" refers to the parameter settings in Table 1. "#B" represents the number of branches. "#D" represents the dilation rates. The number of branches and dilation rates in the unmentioned stages are set according to the default configuration. The best methods are in bold. SAFE: Scale-adaptive feature extraction; SOD: salient object detection; MAE: mean absolute error.

work. The module is mainly divided into multi-scale feature interaction and dynamic selection. Multi-scale feature interaction is used to realize cross-scale feature embedding and improve the representation ability of the network within the layer; features of various scales have different representation abilities for salient targets. To measure this difference, we deploy dynamic selection after multi-scale feature interaction to extract useful information by assigning different weights to features of different scales. We complete the design of the backbone network with the SAFE module as the basic unit and combine it with a decoder based on the MFA module to realize the final SANet. We use four quantitative metrics, maxF, avgF, MAE, and S, to evaluate the effectiveness of the model on six commonly used SOD datasets and a traffic dataset, and use parameters (#Param), FLOPs, and FPS to evaluate the effectiveness. In addition, SANet is qualitatively compared with state-of-the-art heavyweight and lightweight methods. The final results show that SANet achieves 62 fps on an NVIDIA GTX 3090 GPU with only 2.29 M parameters, significantly outperforming other models. In terms of model performance, it matches the performance of general heavyweight methods and surpasses three other state-of-the-art lightweight methods.

In this paper, we have conducted extensive ablation experiments to validate the parameter selection of the SAFE module, although further research on its theoretical foundation is needed. Therefore, in future work, we will further explore this theoretical basis. Additionally, we aim to improve the detection performance of the proposed model and expand its applicability to more scenarios.

DECLARATIONS

Authors' contributions

Made substantial contributions to conception and design of the study, performed data analysis and interpretation, and wrote the manuscript: Liu Z

Provided technical guidance and reviewed the manuscript: Zhao W, Jia N
Reviewed the manuscript: Liu X, Yang J

Availability of data and materials

The raw data that support the findings of this study are available from the corresponding author Zhao W, upon reasonable request. The datasets used in the paper and other model-related image data have been appropriately cited in References with the standard referencing conventions.

Financial support and sponsorship

This work was supported by the National Natural Science Foundation of China (62403361).

Conflicts of interest

All authors declared that there are no conflicts of interest.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Copyright

REFERENCES

1. Jia N, Chen Y, Liu X, Wang H. DIG: dual interaction and guidance network for salient object detection. *Appl Intell* 2023;53:28039–53. DOI
2. Itti L, Koch C, Niebur E. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans Pattern Anal Mach Intell* 1998;20:1254–9. DOI
3. Liu T, Yuan Z, Sun J, et al. Learning to detect a salient object. *IEEE Trans Pattern Anal Mach Intell* 2010;33:353–67. DOI
4. Tang G, Ni J, Shi P, Li Y, Zhu J. An improved ViBe-based approach for moving object detection. *Intell Robot* 2022;2:130-44. DOI
5. Geng Y, Lian Y, Yang S, Zhou M, Cao J. Reliable part guided multiple level attention learning for person re-identification. *J Circuits Syst Comput* 2021;30:2150246. DOI
6. Ji A, Woo WL, Wong EWL, Quek YT. Rail track condition monitoring: a review on deep learning approaches. *Intell Robot* 2021;1:151–75. DOI
7. Zhao KWG, Wang Y, Ma S, Lu J. SaliencyVR: saliency matching based visual relocalization for autonomous vehicle. *IEEE Trans Intell Veh* 2024:1-10. DOI
8. Li X, Zhang T, Liu Z, et al. Saliency guided siamese attention network for infrared ship target tracking. *IEEE Trans Intell Veh* 2024:1-18. DOI
9. Ding N, Zhang C, Eskandarian A. SalienDet: a saliency-based feature enhancement algorithm for object detection for autonomous driving. *IEEE Trans Intell Veh* 2023;9:2624–35. DOI
10. Qin L, Shi Y, He Y, et al. ID-YOLO: real-time salient object detection based on the driver's fixation region. *IEEE Trans Intell Transp Syst* 2022;23:15898–908. DOI
11. Qian W, He Z, Chen C, Peng S. Navigating diverse salient features for vehicle re-identification. *IEEE Trans Intell Transp Syst* 2022;23:24578–87. DOI
12. Ravindran R, Santora MJ, Jamali MM. Multi-object detection and tracking, based on DNN, for autonomous vehicles: a review. *IEEE Sens J* 2020;21:5668–77. DOI
13. Liu N, Han J, Yang MH. Picanet: learning pixel-wise contextual attention for saliency detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2018. pp. 3089–98. DOI
14. Pang Y, Zhao X, Zhang L, Lu H. Multi-scale interactive network for salient object detection. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020 Jun 13-19; Seattle, USA. IEEE; 2020. pp. 9413–22. DOI
15. Cheng MM, Gao SH, Borji A, Tan YQ, Lin Z, Wang M. A highly efficient model to study the semantics of salient object detection. *IEEE Trans Pattern Anal Mach Intell* 2021;44:8006–21. DOI
16. Liu Y, Gu YC, Zhang XY, Wang W, Cheng MM. Lightweight salient object detection via hierarchical visual perception learning. *IEEE Trans Cybern* 2020;51:4439–49. DOI
17. Liu Y, Zhang XY, Bian JW, Zhang L, Cheng MM. SAMNet: stereoscopically attentive multi-scale network for lightweight salient object detection. *IEEE Trans Image Process* 2021;30:3804–14. DOI

18. Howard AG, Zhu M, Chen B, et al. Mobilenets: efficient convolutional neural networks for mobile vision applications. *arXiv* 2017;arXiv:1704.01861. Available from: https://doi.org/10.48550/arXiv.1704.04861. [accessed 23 December 2024].
19. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC. Mobilenetv2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2018. pp. 4510–20. Available from: https://openaccess.thecvf.com/content_cvpr_2018/html/Sandler_MobileNetV2_Inverted_Residuals_CVPR_2018_paper.html. [accessed 23 December 2024].
20. Zhang X, Zhou X, Lin M, Sun J. Shufflenet: an extremely efficient convolutional neural network for mobile devices. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2018 Jun 18-23; Salt Lake City, USA. IEEE; 2018. pp. 6848–56. DOI
21. Ma N, Zhang X, Zheng HT, Sun J. ShuffleNet V2: practical guidelines for efficient CNN architecture design. In: Proceedings of the European conference on computer vision (ECCV); 2018. pp. 122-38. DOI
22. Jia N, Sun Y, Liu X. TFGNet: traffic salient object detection using a feature deep interaction and guidance fusion. *IEEE Trans Intell Transp Syst* 2023;25:3020-30. DOI
23. Cheng MM, Mitra NJ, Huang X, Torr PH, Hu SM. Global contrast based salient region detection. *IEEE Trans Pattern Anal Mach Intell* 2014;37:569–82. DOI
24. Wang L, Lu H, Ruan X, Yang MH. Deep networks for saliency detection via local estimation and global search. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2015 Jun 07-12; Boston, USA. IEEE; 2015. pp. 3183–92. DOI
25. Zhang P, Wang D, Lu H, Wang H, Ruan X. Amulet: aggregating multi-level convolutional features for salient object detection. In: 2017 IEEE International Conference on Computer Vision (ICCV); 2017 Oct 22-29; Venice, Italy. IEEE; 2017. pp. 202–11. DOI
26. Chen Z, Xu Q, Cong R, Huang Q. Global context-aware progressive aggregation network for salient object detection. *Proc AAAI conf Artif Intell* 2020;34:10599–606. DOI
27. Huang K, Tian C, Su J, Lin JCW. Transformer-based cross reference network for video salient object detection. *Pattern Recogn Lett* 2022;160:122–7. DOI
28. Li Y, Ma J. A swin transformer-based asymmetrical network for light field salient object detection. In: 5th International Conference on Information Science, Electrical, and Automation Engineering (ISEAE 2023). 2023. pp. 171–6. DOI
29. Han S, Pool J, Tran J, Dally WJ. Learning both weights and connections for efficient neural network. In: Proceedings of the 28th International Conference on Neural Information Processing Systems. 2015. pp. 1135-43. Available from: https://dl.acm.org/doi/10.5555/2969239.2969366. [Last accessed on 23 Dec 2024]
30. Courbariaux M, Bengio Y, David JP. BinaryConnect: training deep neural networks with binary weights during propagations. In: Proceedings of the 28th International Conference on Neural Information Processing Systems. 2015. pp. 3123-31. Available from: https://dl.acm.org/doi/10.5555/2969442.2969588. [Last accessed on 23 Dec 2024]
31. Huang Z, Wang N. Like what you like: knowledge distill via neuron selectivity transfer. *arXiv* 2017;arXiv:1707.01219. Available from: https://doi.org/10.48550/arXiv.1707.01219. [accessed 23 December 2024].
32. Tan M, Le Q. Efficientnet: rethinking model scaling for convolutional neural networks. In: Proceedings of the 36th International Conference on Machine Learning. PMLR; 2019. pp. 6105–14. Available from: https://proceedings.mlr.press/v97/tan19a.html?ref=jina-ai-gmbh.ghost.io. [Last accessed on 23 Dec 2024].
33. Tan M, Le Q. Efficientnetv2: smaller models and faster training. In: Proceedings of the 36th International Conference on Machine Learning. PMLR; 2021. pp. 10096–106. Available from: https://proceedings.mlr.press/v139/tan21a.html. [Last accessed on 23 Dec 2024].
34. Han K, Wang Y, Tian Q, Guo J, Xu C, Xu C. Ghostnet: more features from cheap operations. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020 Jun 13-19; Seattle, USA. IEEE; 2020. pp. 1580–9. DOI
35. Wang Z, Zhang Y, Liu Y, Zhu D, Coleman SA, Kerr D. Elwnet: an extremely lightweight approach for real-time salient object detection. *IEEE Trans Circuits Syst Video Technol* 2023;33:6404-17. DOI
36. Sreelakshmi K, Akarsh S, Vinayakumar R, Soman KP. Capsule neural networks and visualization for segregation of plastic and non-plastic wastes. In: 2019 5th international conference on advanced computing & communication systems (ICACCS); 2019 Mar 15-16; Coimbatore, India. IEEE; 2019. pp. 631–6. DOI
37. Hinton GE, Sabour S, Frosst N. Matrix capsules with EM routing. In: International conference on learning representations; 2018. Available from: https://openreview.net/forum?id=HJWLfGWRb&. [Last accessed on 23 Dec 2024].
38. Saqur R, Vivona S. Capsgan: using dynamic routing for generative adversarial networks. In: Advances in Computer Vision: Proceedings of the 2019 Computer Vision Conference (CVC). Springer; 2020. pp. 511–25. DOI
39. Cheng X, He J, He J, Xu H. Cv-CapsNet: complex-valued capsule network. *IEEE Access* 2019;7:85492–9. DOI
40. Sun K, Yuan L, Xu H, Wen X. Deep tensor capsule network. *IEEE Access* 2020;8:96920–33. DOI
41. Liu Y, Dong X, Zhang D, Xu S. Deep unsupervised part-whole relational visual saliency. *Neurocomputing* 2024;563:126916. DOI
42. Zhang Q, Duanmu M, Luo Y, Liu Y, Han J. Engaging part-whole hierarchies and contrast cues for salient object detection. *IEEE Trans Circuits Syst Video Technol* 2021;32:3644–58. DOI
43. Liu Y, Zhou L, Wu G, Xu S, Han J. Tcgnet: type-correlation guidance for salient object detection. *IEEE Trans Intell Transp Syst* 2023;25:6633-44. DOI
44. Zhao H, Shi J, Qi X, Wang X, Jia J. Pyramid scene parsing network. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2017 Jul 21-26; Honolulu, USA. IEEE; 2017. pp. 2881–90. DOI
45. Wang L, Lu H, Wang Y, Feng M, Wang D, Yin B. Learning to detect salient objects with image-level supervision. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2017 Jul 21-26; Honolulu, USA. IEEE; 2017. pp. 136–45. DOI
46. Yang C, Zhang L, Lu H, Ruan X, Yang MH. Saliency detection via graph-based manifold ranking. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition; 2013 Jun 23-28; Portland, USA. IEEE; 2013. pp. 3166–73. DOI

47. Yan Q, Xu L, Shi J, Jia J. Hierarchical saliency detection. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition; 2013 Jun 23-28; Portland, USA. IEEE; 2013. pp. 1155–62. DOI
48. Li Y, Hou X, Koch C, Rehg JM, Yuille AL. The secrets of salient object segmentation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition; 2014 Jun 23-28; Columbus, USA. IEEE; 2014. pp. 280–7. DOI
49. Li G, Yu Y. Visual saliency based on multiscale deep features. *IEEE Trans Image Process* 2016;25:5012-24. DOI
50. Movahedi V, Elder JH. Design and perceptual validation of performance measures for salient object segmentation. In: 2010 IEEE computer society conference on computer vision and pattern recognition-workshops; 2010 Jun 13-18; San Francisco, USA. IEEE; 2010. pp. 49–56. DOI
51. Wu Z, Su L, Huang Q. Cascaded partial decoder for fast and accurate salient object detection. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2019 Jun 15-20; Long Beach, USA. IEEE; 2019. pp. 3907–16. DOI
52. Qin X, Zhang Z, Huang C, Dehghan M, Zaiane OR, Jagersand M. U^2-Net: going deeper with nested U-structure for salient object detection. *Pattern Recognit* 2020;106:107404. DOI
53. Zhang P, Wang D, Lu H, Wang H, Yin B. Learning uncertain convolutional features for accurate saliency detection. In: 2017 IEEE International Conference on Computer Vision (ICCV); 2017 Oct 22-29; Venice, Italy. IEEE; 2017. pp. 212–21. DOI
54. Hou Q, Cheng MM, Hu X, Borji A, Tu Z, Torr PHS. Deeply supervised salient object detection with short connections. *IEEE Trans Pattern Anal Mach Intell* 2019;41:815-28. DOI
55. Qin X, Zhang Z, Huang C, Gao C, Dehghan M, Jagersand M. BASNet: boundary-aware salient object detection. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2019 Jun 15-20; Long Beach, USA. IEEE; 2019. pp. 7479–89. DOI
56. Liu JJ, Hou Q, Cheng MM, Feng J, Jiang J. A simple pooling-based design for real-time salient object detection. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2019 Jun 15-20; Long Beach, USA. IEEE; 2019. pp. 3917–26. DOI
57. Liu N, Zhang N, Wan K, Shao L, Han J. Visual saliency transformer. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV); 2021 Oct 10-17; Montreal, Canada. IEEE; 2021. pp. 4722–32. DOI
58. Ma M, Xia C, Li J. Pyramidal feature shrinking for salient object detection. In: Proceedings of the AAAI conference on artificial intelligence. 2021. pp. 2311–8. DOI
59. Zhuge M, Fan DP, Liu N, Zhang D, Xu D, Shao L. Salient object detection via integrity learning. *IEEE Trans Pattern Anal Mach Intell* 2022;45:3738–52. DOI
60. Wang Y, Wang R, Fan X, Wang T, He X. Pixels, regions, and objects: multiple enhancement for salient object detection. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2023 Jun 17-24; Vancouver, Canada. IEEE; 2023. pp. 10031–40. DOI
61. Han C, Li G, Liu Z. Two-stage edge reuse network for salient object detection of strip steel surface defects. *IEEE Trans Instrum Meas* 2022;71:1–12. DOI
62. Cui W, Song K, Feng H, Jia X, Liu S, Yan Y. Autocorrelation aware aggregation network for salient object detection of strip steel surface defects. *IEEE Trans Instrum Meas* 2023;72:1-12. DOI