Computers and Electrical Engineering 118 (2024) 109296

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

A Two-branch Edge Guided Lightweight Network for infrared image saliency detection

Zhaoying Liu [a], Xiang Li [a], Ting Zhang [a], Xuesi Zhang [a], Changming Sun [b], Sadaqat ur Rehman [c,*], Jawad Ahmad [d]

[a] Facult of Information Technology, Beijing University of Technology, NO.100 Pingleyuan, Chaoyang District, 100124, Beijing, China
[b] Data61, CSIRO, PO Box 76, Epping, 1710, NSW, Australia
[c] School of Sciences, Engineering and Environment, University of Salford, Manchester, M5 4WT, UK
[d] School of Computing, Engineering and the Built Environment, Edinburgh Napier University, Edinburgh, EH10 5DT, UK

ARTICLE INFO

Keywords:
Infrared images
Lightweight module
Saliency detection
Contour information
Feature integration

ABSTRACT

In the dynamic landscape of saliency detection, convolutional neural networks have emerged as catalysts for innovation, but remain largely tailored for RGB imagery, falling short in the context of infrared images, particularly in memory-restricted environments. These existing approaches tend to overlook the wealth of contour information vital for a nuanced analysis of infrared images. Addressing this notable gap, we introduce the novel Two-branch Edge Guided Lightweight Network (TBENet), designed explicitly for the robust analysis of infrared image saliency detection. The main contributions of this paper are as follows. First, we formulate the saliency detection task as two subtasks, contour enhancement and foreground segmentation. Therefore, the TBENet is divided into two specialized branches: a contour prediction branch for extracting target contour and a saliency map generation branch for separating the foreground from the background. The first branch employs an encoder–decoder architecture to meticulously delineate object contours, serving as a guiding blueprint for the second branch. This latter segment adeptly integrates spatial and semantic data, creating a precise saliency map that is refined further by an innovative edge-weighted contour loss function. Second, to enhance feature integration capabilities, we propose depthwise multi-scale and multi-cue modules, facilitating sophisticated feature aggregation. Third, a high-level linear bottleneck module is devised to ensure the extraction of rich semantic information, and by replacing the standard convolution with the depthwise convolution, it is beneficial to reduce model complexity. Additional, we reduce the number of channels of the feature maps from each stage of the decoder to further enhance the lightweight of the model. Last, we construct a novel infrared ship dataset Small-IRShip to train and evaluate our proposed model. Experimental results on the homemade dataset Small-IRShip and two publicly available datasets, namely RGB-T and IRSTD-1k, demonstrate TBENet's superior performance over state-of-the-art methods, affirming its effectiveness in harnessing edge information and incorporating advanced feature integration strategies.

* Corresponding author.
E-mail addresses: zhaoying.liu@bjut.edu.cn (Z. Liu), lixiang0123@emails.bjut.edu.cn (X. Li), zhangting@bjut.edu.cn (T. Zhang), gan_enyu111@163.com (X. Zhang), changming.sun@csiro.au (C. Sun), s.rehman15@salford.ac.uk (S.u. Rehman), j.ahmad@napier.ac.uk (J. Ahmad).

https://doi.org/10.1016/j.compeleceng.2024.109296
Received 25 December 2023; Received in revised form 17 April 2024; Accepted 6 May 2024
Available online 28 May 2024
0045-7906/

1. Introduction

Salient object detection (SOD) is a category-agnostic task that focuses on segmenting salient targets, regardless of their specific category [1]. Consequently, SOD is extensively used to support various computer vision applications, including segmentation [2–4], classification [5–7], recognition [8–10], and tracking [11,12]. Most developed SOD models are trained on visible light images because the RGB images can provide a visually appealing experience, allowing for easy identification of object categories through clear color and appearance features [13]. However, they struggle in dark environments [14]. Recently, as hardware devices such as thermal sensors continue to be upgraded, infrared (IR) imaging systems are becoming less expensive and the quality of the imaging is becoming better, which makes infrared saliency detection algorithms [15] attract more attention. In contrast to visible light images, IR images thrive in dark environments and excel at long-distance images [16,17]. Therefore, IR image saliency detection is better suited for marine [18] and military applications [19], such as maritime search and rescue, ship detection, and related fields [20].

With the growing popularity of vision technology [21,22], numerous SOD algorithms have emerged [23,24]. These approaches can be broadly categorized into traditional methods and deep learning (DL) based methods [25,26]. Traditional methods heavily rely on prior knowledge for saliency analysis. For instance, Zhu et al. [27] utilize boundary connectivity as a robust background measure to differentiate image patches. Liu et al. [28] introduce contrast priori and combines local, regional, and global features using a conditional random field. Yang et al. [29] compute similarity between image elements and queries, ranking them with a graph-based manifold. However, traditional SOD methods struggle with challenging environments and background clutter. Fortunately, DL-based methods address these limitations by leveraging deep nonlinear networks to extract features, resulting in improved accuracy. Hou et al. [30] introduce short connections in a deeply supervised network for SOD. Zhang et al. [31] enable message passing between layers using a gated bi-directional module to fuse multi-scale feature maps. In our previous work [32], we propose DG-Light-NLDF, which incorporates a global information extraction module and a dilated linear bottleneck to reduce the number of model parameters.

While DL-based saliency detection methods have made significant progress [33], there is still room for improvement [34]. Firstly, many SOD methods employ complex modules with a large number of parameters, which is not suitable for infrared images. Lightweight modules, such as depthwise separable convolution [35], ghost features [36], and micro-factorized convolution [37], have shown promising results with limited computational power but are yet to be explored in SOD. Secondly, the utilization of edge information in existing frameworks remains an area of interest. Several models [38–41] have incorporated edge information to improve accuracy and guide saliency map production. However, these models treat each sample equally, which can make them fragile [42]. In summary, further advancements in SOD can be achieved by exploring lightweight modules suitable for infrared images and by better leveraging edge information.

In this paper, we propose a two-branch edge guided lightweight saliency detection model specifically designed for infrared images. We aim to enhance the performance of saliency detection by improving the accuracy of edge detection and devise some lightweight modules to reduce the complexity of the model. Due to the inconspicuous difference between the target and the background in infrared scenes [43,44], which results in blurred edges, and the fact that most of the infrared cameras are mounted on edge devices, which require high efficiency [45,46]. We proposed efficient network for generating saliency maps with accurate edges can better address the saliency analysis of infrared images. To this end, our two-branch network contains a contour prediction branch and a saliency map generation branch. The former is responsible for extracting the contour information of the target. The latter in charge of computing a uniform saliency map guided by the contour information. Unlike other edge-based SOD approaches that employ complex structures or specific edge loss functions [47], we propose an edge-weighted contour loss for supervising the training of the saliency map generation branch. This loss assigns a higher weight to samples located in edge regions, enabling our network to focus more on regional loss on edges during training. Additionally, we introduce a depthwise multi-scale integration module to fuse features with different resolutions for restoring spatial information of targets and a multi-clue integration module to combine low-level, mid-level, and high-level features for thoroughly representing targets. Furthermore, in order to capture rich semantic features while maintaining high efficiency, we construct a high-level linear bottleneck module that replaces standard convolution with depthwise convolution. And we reduce the channel numbers of the feature maps in the decoder to further decrease model complexity. Finally, to address the lack of publicly available infrared datasets, we construct a dataset called Small-IRShip, which consists of 1002 infrared ship images with corresponding edge labels and saliency labels. Two additional datasets, RGB-T [48] and IRSTD-1k [49], are selected to validate the generalization performance of the TBENet. Experimental results on these three datasets demonstrate that the TBENet outperforms state-of-the-art methods. Our contributions are outlined as:

1. We propose a two-branch edge guided lightweight network (TBENet) for saliency detection in infrared images. It consists of a contour prediction branch that captures edge information and a saliency map generation branch that produces clear saliency maps. Additionally, we introduce a contour loss to the final loss function to enhance edge prediction accuracy.
2. Our TBENet incorporates two feature integration modules: a depthwise multi-scale integration module and a multi-clue integration module, which enable the extraction of more refined features.
3. We design a lightweight high-level linear bottleneck module and reduce the number of channels of feature maps output by each stage in the decoder to reduce the number of parameters without compromising detection accuracy.
4. To train our model and provide a benchmark for evaluation, we construct a novel dataset called Small-IRShip. Extensive experiments are conducted on this dataset to assess the performance of our detector.

The remainder of this paper is organized as follows: Section 2 discusses related studies on salient object detection. Section 3 presents a detailed description of the overall structure of TBENet. Section 4 showcases quantitative and qualitative experimental results. Finally, Section 5 concludes the paper.

2. Related work

The increasing popularity of convolutional neural networks (CNNs) [50,51] has led to the widespread adoption of fully convolutional neural networks (FCNs) [52,53] in SOD due to their ability to preserve spatial information. In FCN-based networks, features at different levels play different roles due to their varying receptive fields [54]. However, effectively integrating these features poses a challenge for SOD [55]. Many SOD methods employ multi-feature fusion techniques [56]. Ge et al. [57] use a group semantic module to guide multi-layers feature fusion to learn the consistent and discriminative co-salient features. Ren et al. [58] integrate multi-scale features of different modalities via a mask-guided feature aggregation module. Huang et al. [59] investigate the aggregation and propagation of features across deep CNN layers. Luo et al. [60] use cascaded sub-modules to capture and integrate multiple resolution feature maps effectively. Liu et al. [61] explore the potential of pooling layers in a global feature guidance module and a feature fusion module to enrich saliency maps. Zhao et al. [62] separately handle semantic context, spatial details, and boundary information in the decoder component and progressively merges these features using three specific integration modules. Song et al. [63] use a transformer-based backbone and a convolution-based backbone to extract features, respectively. Then, a hybrid attention mechanism is utilized to fuse the features from different backbones. Yan et al. [64] propose a four-branch feature integration module to fuse feature maps at different scales. Each branch has a different receptive field. To address the dilution of high-level features, Chen et al. [65] incorporate attention mechanisms to enhance the top layer features and integrates features guided by contextual information. Chen et al. [66] integrate semantic, spatial, and global context information using a similarity fusion module to enhance the complementary of maps. Li et al. [67] insert the attention mechanism into the convolutional layer in order to extract useful global information. Sun et al. [68] devise a cross-modality feature dynamic fusion module that generates different weights for different modalities for complementarily conducting RGB-T features fusion. Huang et al. [69] propose a multi-scale saliency detection model that takes images at three different scales and uses the saliency maps of neighboring scales to guide the current features fusion.

Considering the benefits of incorporating edge information for accurate saliency maps [70], several studies [71] have focused on embedding boundary clues in SOD approaches [72]. Luo et al. [38] propose an effective network for saliency detection (NLDF) that extracts multi-scale features through a grid-shaped multi-resolution network, enhances feature contrast using a contrast module, and optimizes boundaries using a boundary loss function. Chen et al. [73] introduce an edge-aware refinement module in the decoder to segment objects with clear boundaries. Wu et al. [74] develop a cross refinement unit that exchanges messages between saliency prediction and boundary detection tasks, enabling the simultaneous generation of precise saliency maps and edge maps. In the primary network, Han et al. [75] obtain an edge map in the encoder and uses it to guide the initial saliency map in the decoder. The edge map is further incorporated into the sub-network to refine the initial saliency map. Zheng et al. [76] use two independent decoders with shared encoder for generating smooth edges and accurate saliency maps by fusing three deep feature maps, respectively. In SDFNet [77], the differences between the different channels in the RGB color space allow the network to extract the edge information of the target. And then, the edge information and the original image are fed into the Siamese network for learning the complementary cross-modal information between them. Zhang et al. [78] utilize the extracted edge cues to compute spatial attention weights and apply them to the feature maps for spatial feature selection, facilitating the network to perceive edge details. Zeng et al. [79] propose a difference perception mechanism that extracts the edge details of the target by calculating the difference between the max pooling and the average pooling. Zhou et al. [80] apply the predicted edge maps to multiple saliency prediction encoders, and manipulate the encoders to produce saliency maps with smooth edges by fusing the edge maps with multi-scale feature maps.

In recent years, lightweight modules [81] have garnered significant attention due to the complex structures and large parameter numbers of previous models [82]. Zhang et al. [83] utilize channel cleaning and point-wise convolution to reduce computational complexity and improve results, addressing the lack of information exchange between different channels of group convolutions. Iandola et al. [84] employ compression and expansion operations to construct lightweight modules and reduce model complexity. Gao et al. [85] introduce the concept of channel convolution, which sparsely connects input and output channels in a sliding window-like fashion to maintain information exchange between channels. Sandler et al. [86] introduce an inverse residual linear bottleneck module, which outperforms the depthwise separable convolution module by addressing computation-related issues. Zhang et al. [87] decompose ordinary convolutions into multiple group convolutions and employs a sorting module to enable information circulation between channels. Xie et al. [88] propose a cross-structured channel convolution to resolve the dense group convolution channel problem. While lightweight modules can significantly enhance computational efficiency and reduce the number of parameters, they have been rarely utilized in salient object detection. Leveraging the specific characteristics of infrared images, Liu et al. [89] propose a lightweight NLDF model by simplifying the feature extraction and fusion modules. Liu et al. [32] replace the original convolution module with a dilated linear bottleneck module and achieves outstanding performance under the guidance of a global feature extraction module. However, the edge information is not considered. Therefore, we take edge information and lightweight module into consideration simultaneously to improve the performance of our detector.

3. The proposed method

3.1. Overview

In this section, we present the overall structure of our proposed two-branch edge guided lightweight network (TBENet), as shown in Fig. 1. Our network includes an encoder, a decoder, a contour prediction branch, and a saliency map generation branch. The input

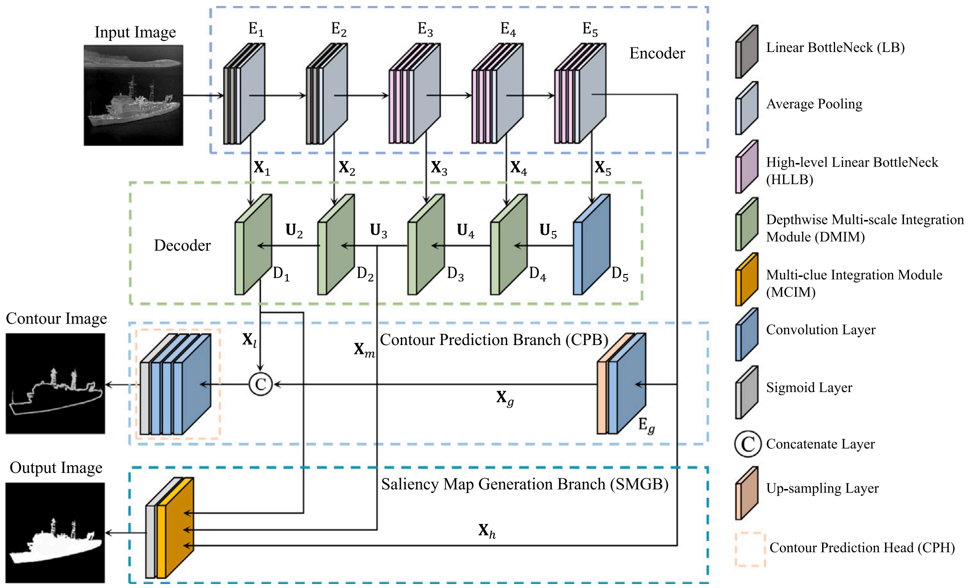

Fig. 1. The entire structure of TBENet. Our model is composed of four parts: an encoder, a decoder, a contour prediction branch, and a saliency map generation branch. The encoder–decoder network progressively produces fine-grained features. Given the refined features, the contour prediction branch delineates clear target contour. Guided by the contour information, the saliency map generation branch aggregates multi-dimensional information to obtain high quality saliency map.

of the TBENet is an infrared image and the output of the TBENet is an image pair consisting of a contour image and a saliency map. The encoder first extracts multi-scale feature maps denoted by $\{\mathbf{X}_i\}_{i=1}^{5}$ through five sub-modules denoted by $\{\mathrm{E}_i\}_{i=1}^{5}$. And then, the decoder merges multi-scale feature maps via a series of sub-modules denoted by $\{\mathrm{D}_i\}_{i=1}^{5}$, generating multiple intermediate refined features $\{\mathbf{X}_l, \{\mathbf{U}_i\}_{i=2}^{5}\}$. After that, the contour prediction branch captures the global information $\mathbf{X}_g$ obtained by a module E_g. The $\mathbf{X}_g$ and the enhanced feature $\mathbf{X}_l$ are integrated to establish the contour information of the targets. The saliency map generation branch produces the final saliency map by fusing the contour information with the mid-level spatial features $\mathbf{X}_m$ and the high-level semantic features $\mathbf{X}_h$ using a multi-clue integration module. Under the guidance of the contour information, our model can obtain more accurate saliency map with clear edges.

3.2. Encoder

These FCN-based methods [90] typically downsample the input image several times and restore the resolution of the image with only one upsampling operations. This simple technique loses the spatial information of the image. Therefore, we adopt a more advanced framework based on U-Net [91], preserving the spatial information of the image. It contains an encoder and a decoder. The encoder acts as a feature extractor. The decoder plays the role of fusing multi-scale features. We select the VGG16 as the backbone of the encoder. Shallow features contain structural information about the target. However, standard convolution involves more channel-level operations that interfere with the extraction of structural features. Therefore, we replace the standard convolution with linear bottleneck (LB) [92]. The LB contains depthwise convolution operations that reduce inter-channel interactions and improve the extraction of structural features while reducing network complexity. The LB is introduced into the first two sub-modules, namely E_1 and E_2. The structure of the LB is shown in Fig. 2(a). Specifically, given an input $\mathbf{X} \in \mathbb{R}^{W\times H\times C}$, we use a pointwise convolution (PW) to transform the features, obtaining $\mathbf{X}_1 \in \mathbb{R}^{W\times H\times C}$. For decreasing the number of parameters, a 3 × 3 depthwise convolution (DW) is leveraged to replace the original convolution, generating $\mathbf{X}_2 \in \mathbb{R}^{W\times H\times C}$. The $\mathbf{X}_3 \in \mathbb{R}^{W\times H\times C'}$ is the output by a PW. Deep features contain semantic information. In order to obtain informative semantic features, we design a high-level linear bottleneck module (HLLB) that employs various convolution kernels to expand the respective field and enhance the extraction of global semantic information. The details of the HLLB is shown in Fig. 2(b). Concretely, the HLLB has three layers. The first layer is a PW with input $\mathbf{X}$, producing $\mathbf{X}_f$. Differing from the common LB, the second layer consists of three DWs with the kernel sizes of 3, 5, and 7, respectively. The $\mathbf{X}_f$ is fed into different DW separately to obtain three feature maps with various receptive fields, $\mathbf{X}_{s3}$, $\mathbf{X}_{s5}$, and $\mathbf{X}_{s7}$. Concatenation is then performed for merging the three feature maps. The third layer is still a PW that reduces channel dimensions and integrates features. Finally, similar to residual neural network, a shortcut connection is applied between the input and output to obtain the final feature map $\mathbf{X}_o$. The above process is as follows:

$$
\begin{aligned}
\mathbf{X}_f &= PW(\mathbf{X}), \\
\mathbf{X}_o &= PW(Cat(DW_3(\mathbf{X}_f), DW_5(\mathbf{X}_f), DW_7(\mathbf{X}_f))) + \mathbf{X},
\end{aligned}
\tag{1}
$$

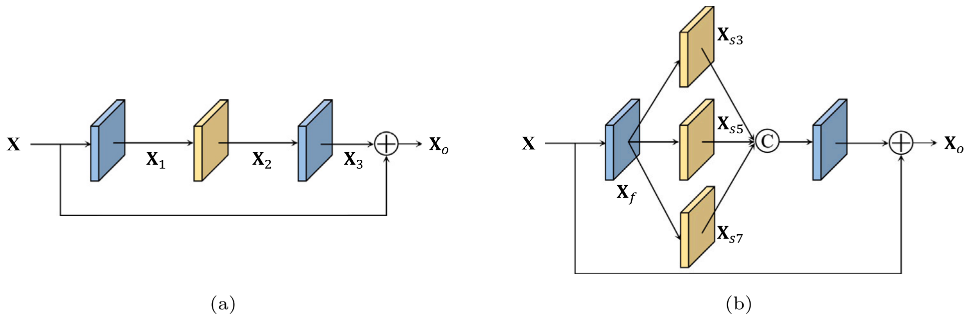

Fig. 2. The structure of the different linear bottleneck modules. (a) means the common linear bottleneck module; (b) denotes the proposed high-level linear bottleneck module (HLLB). The sub-modules marked in yellow and blue are the depthwise convolution and the standard convolution. $\oplus$ stands for an elementwise addition layer.

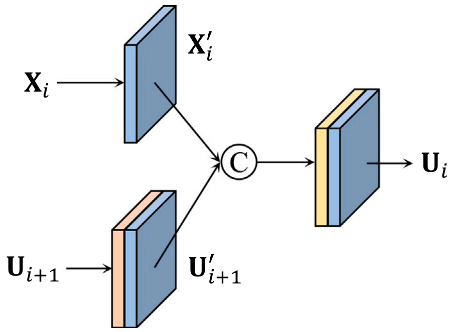

Fig. 3. The details of the depthwise multi-scale integration module (DMIM). The sub-module marked in pink stands for a up-sampling layer.

where Cat is the concatenate operation along channel. DW_3, DW_5, and DW_7 mean the 3×3, 5×5, and 7×7 depthwise convolutions, respectively. LB enriches the structural information and HLLB boosts the semantic information, which jointly enhance the representation of the feature extraction network and provide representative feature maps for the saliency detection task.

3.3. Decoder

In order to recover the spatial information of the target, we divide the decoder into four upsampling stages. In each stage, the coarse semantic information from the high level is first upsampled to recover the resolution, and afterwards, it is integrated with the feature maps containing detailed information from the low level to reduce the loss of spatial information caused by upsampling. However, utilizing the original convolution to integrate the features results in a large number of parameters and unsatisfactory performance. To remedy this problem, we propose a simple depthwise multi-scale integration module (DMIM), as shown in Fig. 3. Our decoder consists of a convolution layer D_5 and four consecutive DMIMs denote by $\{D_i\}_{i=1}^{4}$. The experimental results given in Section 4 demonstrate that the DMIM not only achieves ideal feature integration effects but also has a small number of parameters.

We first apply D_5 with a kernel size of 3×3 to adjust the deep level features $\mathbf{X}_5$. Then, the adjusted features pass through four DMIMs to obtain the advanced features. For convenience, the ith DMIM is used as an example to explain our method. Notably, this DMIM has two inputs: low-level feature $\{\mathbf{X}_i\}_{i=1}^{4}$ output by the corresponding layer from the encoder and high-level feature $\mathbf{U}_{i+1}$ generated from the previous module. First, we use a PW to reduce the channel dimension of $\mathbf{X}_i$, containing $\mathbf{X}'_i$. Then $\mathbf{U}_{i+1}$ is processed by an up-sampling layer to resize it to the same size as $\mathbf{X}_i$, which is followed by a PW to contain $\mathbf{U}'_{i+1}$. Subsequently, we concatenate $\mathbf{X}'_i$ and $\mathbf{U}'_{i+1}$ along the channel dimension. The result is fed into a 3×3 DW for reducing the number of parameters, which followed by a PW to produce the final output $\mathbf{U}_i$. The process is summarized in Eq. (2).

$$\mathbf{U}_i = PW(DW(Cat(PW(f_u(\mathbf{U}_{i+1})), PW(\mathbf{X}_i)))), \tag{2}$$

where f_u stands for an up-sampling layer implemented by a bilinear interpolation operation. The feature map from the last layer of the encoder is passed through multiple DMIMs and the spatial structure of the target is recovered, making it easier for the network to discriminate the state (foreground or background) of each pixel location and facilitating the generation of a clear saliency map.

To further enhance the lightweight of our model, we reduce the number of channels of feature maps from each DMIM. In our previous work [32], the number of channels of the output feature map for each upsampling step is equal to half of the sum of the number of channels of the two input feature maps. Since infrared image features are relatively simple and too many channel numbers tend to cause overfitting, we believe that the number of channels in the feature map should be reduced. Therefore, we define the number of output channels as half the number of input channels. Details are shown in Table 1.

Table 1
Details of each stage of the decoder. Original and Reduced denote the feature maps for the number of unreduced and reduced channels, respectively.

D_i	Input1	Input2	Original	Input1	Input2	Reduced
D_1	176 × 176 × 64	88 × 88 × 224	176 × 176 × 144	176 × 176 × 64	88 × 88 × 64	176 × 176 × 32
D_2	88 × 88 × 128	44 × 44 × 320	88 × 88 × 224	88 × 88 × 128	44 × 44 × 128	88 × 88 × 64
D_3	44 × 44 × 256	22 × 22 × 384	44 × 44 × 320	44 × 44 × 256	22 × 22 × 256	44 × 44 × 128
D_4	22 × 22 × 512	11 × 11 × 256	22 × 22 × 384	22 × 22 × 512	11 × 11 × 512	22 × 22 × 256
D_5	11 × 11 × 512	–	11 × 11 × 256	11 × 11 × 512	–	11 × 11 × 512

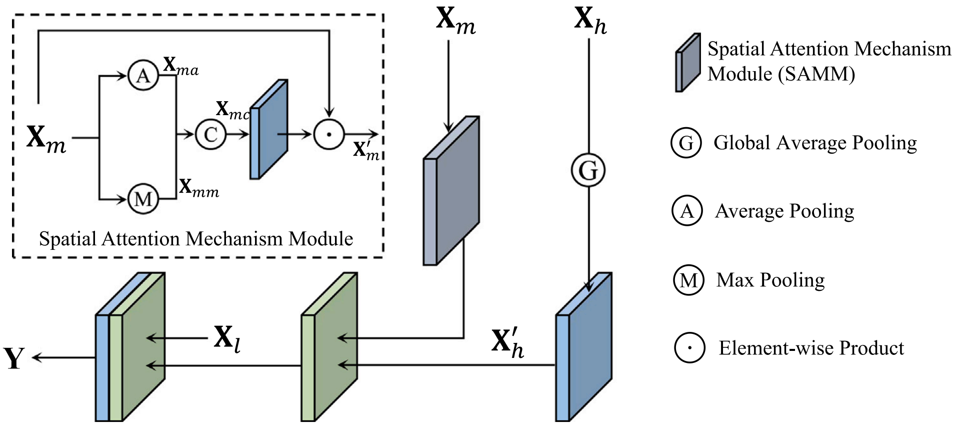

Fig. 4. The structure of the multi-cue integration module (MCIM). The input to the network consists of three parts, i.e., high-level semantic information $\mathbf{X}_h$, mid-level spatial information $\mathbf{X}_m$, and low-level detail information $\mathbf{X}_l$. The MCIM first fuses mid-level features and high-level features, and the result is then fused with low-level features. By integrating information from multiple dimensions of the target, MCIM is able to produce informative features that enhance the detection performance of the model.

3.4. Contour prediction branch

Most of these developed detectors overlook the contour information of the target, resulting in blurred edges. Therefore, our contour prediction branch (CPB) attempts to obtain contour information to improve the accuracy of saliency detection in edge regions. As shown in Fig. 1, the CPB is composed of a module E_g and a contour prediction head (CPH). Specifically, the deep features from E_5 is processed by a 3 × 3 convolution layer and reshaped by an up-sampling operation, obtaining the global clues $\mathbf{X}_g$. Then, $\mathbf{X}_l$ from $\mathbf{U}_1$ and $\mathbf{X}_g$ is concatenated along the channel dimension, which is followed by CPH to generate the contour image. The CPH contains three convolution layers for reducing the number of channels to 1 and a sigmoid layer for computing the probability distribution.

In addition, infrared images have blurred edges, which makes labeling more difficult. Consequently, we select a simple method to generate the groundtruth of the contour images. First, a dilation and an erosion operations are applied to the saliency map. Second, the two results are subtracted to produce the contour label, which is followed by a Gaussian filtering. The detailed process is given by Eq. (3).

$$E = f_g(f_d(S) - f_e(S)) + 1, \tag{3}$$

where E is the obtained contour label and S denotes the saliency map. f_g, f_d, and f_e stand for the Gaussian function, the dilation operation, and the erosion operation, respectively. All kernel sizes are 3 × 3.

3.5. Saliency map generation branch

The saliency map generation branch (SMGB) uses the contour information obtained in the CPB to compute the final saliency map. The SMGB consists of a proposed multi-cue integration module (MCIM) and a sigmoid layer. Features at different levels have different roles. Numerous studies have shown that high-level features contain semantic information, mid-level features contain spatial information, and low-level features contain detailed features. Therefore, the MCIM is utilized to fuse low-level features $\mathbf{X}_l$, mid-level features $\mathbf{X}_m$, and high-level features $\mathbf{X}_h$ to produce an accurate saliency map. The sigmoid layer is responsible for calculating the probability of each pixel point belonging to the foreground. The architecture of the MCIM is presented in Fig. 4.

Specifically, the designed MCIM has three inputs: $\mathbf{X}_l \in \mathbb{R}^{176\times176\times144}$, $\mathbf{X}_m \in \mathbb{R}^{44\times44\times320}$, and $\mathbf{X}_h \in \mathbb{R}^{11\times11\times512}$. First of all, $\mathbf{X}_h$ passes through a global average pooling layer and a PW to extract the global features, obtaining $\mathbf{X}'_h$. Second, to strengthen $\mathbf{X}_m$, we introduce a spatial attention mechanism module (SAMM). In SAMM, $\mathbf{X}_m$ is processed by the average pooling (AP) and max pooling (MP) separately to generate $\mathbf{X}_{ma} \in \mathbb{R}^{44\times44\times1}$ and $\mathbf{X}_{mm} \in \mathbb{R}^{44\times44\times1}$. A concatenate operation is applied on $\mathbf{X}_{ma}$ and $\mathbf{X}_{mm}$, which is followed by a 3×3 convolution operator to obtain $\mathbf{X}_{mc} \in \mathbb{R}^{44\times44\times1}$. The result and $\mathbf{X}_m$ perform the element-wise product to produce the enhanced $\mathbf{X}'_m$. The process is given by Eq. (4).

$$\mathbf{X}'_m = Conv(Cat(AP(\mathbf{X}_m), MP(\mathbf{X}_m))) \odot \mathbf{X}_m, \tag{4}$$

Third, $\mathbf{X}'_h$ and $\mathbf{X}'_m$ are fused by the DMIM, which is followed by another DMIM to merge $\mathbf{X}_l$, obtaining the final output $\mathbf{Y}$. The above process is formulated as follows.

$$\mathbf{Y} = Conv(DMIM(\mathbf{X}_l, DMIM(\mathbf{X}'_m, \mathbf{X}'_h))). \tag{5}$$

And then, the saliency map is predicted by applying a sigmoid layer on $\mathbf{Y}$.

3.6. Loss function

The loss function of the proposed TBENet contains the contour prediction loss L_c, the saliency prediction loss L_b (contour loss), and the structural similarity loss L_s. In saliency detection, the binary cross-entropy loss function L_{bce} is usually used to calculate the difference between the groundtruth and predicted results, as given in Eq. (6).

$$L_{bce} = -\sum_{x,y} \left(Y_{x,y} \log(P_{x,y}) + (1 - Y_{x,y}) \log(1 - P_{x,y}) \right), \tag{6}$$

where $Y_{x,y}$ and $P_{x,y}$ represent the values of the groundtruth and the predicted results at the coordinate position (x, y), respectively. In this paper, L_c is implemented by L_{bce}. The common cross-entropy loss assigns the same weight to each sample. However, those samples located in edge regions are more difficult to distinguish compared to those in the interior. Therefore, we use a weighted cross-entropy loss (contour loss) to assign greater weights to these indistinguishable samples.

$$L_b = -\sum_{x,y} W_{x,y} \left(Y_{x,y} \log(S_{x,y}) + (1 - Y_{x,y}) \log(1 - S_{x,y}) \right), \tag{7}$$

where W denotes the weight matrix, which is equal to E, and E is the contour image.

In addition, the relationship between pixels of an image is also an important factor, but the cross-entropy loss function ignores this factor. Therefore, similar to other work [32], we introduce the image similarity evaluation function, namely structural similarity image measurement (SSIM). The definition of SSIM is given in Eq. (8).

$$L_s = SSIM(Y, S) = \frac{(2\mu_y \mu_s + C_1)(2\sigma_{ys} + C_2)}{({\mu_y}^2 + {\mu_s}^2 + C_1)({\sigma_y}^2 + {\sigma_s}^2 + C_2)}, \tag{8}$$

where μ_y and σ_y are the mean and standard deviation of groundtruth Y, respectively; μ_s and σ_s are the mean and standard deviation of the predicted saliency map S, respectively; and σ_{ys} denotes the covariance of Y and S. Parameters C_1 and C_2 are introduced to avoid a denominator of 0, and they are set to $C_1 = 0.0001$, $C_2 = 0.0009$. The final loss function is given in Eq. (9).

$$L = L_c + (1 - L_s) + L_b \tag{9}$$

3.7. Algorithm process

The training algorithm for the processes described so far is shown in Algorithm 1.

Algorithm 1: Two-branch edge guided lightweight network for infrared image saliency detection

In this algorithm, we empirically set the stopping condition as epoch ≥ 100.

Input: Training set (X, Y, E), $x^k \in X$, $y^k \in Y$, $e^k \in E$, $k = 1, 2, \ldots, N$. x^k, y^k, and e^k represent the image, label, and edge of the kth sample, respectively, and N is the total number of samples.

Output: The optimal weights $\widetilde{W}$ and biases $\widetilde{b}$.

```
Use Xavier to initialize weights W and biases b.
while condition not met do
    As shown in Fig. 1, take X as input, execute the feature extractors E1-E5 in turn, and obtain the corresponding outputs X1-X5;
    Extract global feature Xg through Eg, let Xh=X5;
    Implement the sub-module D5-D1 from the decoder, obtain the corresponding outputs U5-U2 and local feature Xl, let Xm=U3;
    Combine global features Xg and local features Xl to predict edge map Ẽ;
    Integrate features Xl, Xm, and Xh through the multi-clue integration module to obtain the final saliency map Ỹ;
    Compute the loss via Eq. (9), and update weights W and biases b;
end
W̃ = W, b̃ = b.
```

Table 2
The detailed description on the Small-IRShip dataset. Frames and Size denote the number of frames and their total size, respectively. Boxes indicates the number of targets. Large, Medium, and Small denote the number of large, medium, and small targets, respectively.

Class	hzy	kst	ldk	mru	nzd	qtc	rak	rxd	tmk	tqs	xys	zxs
Frames	81	81	86	81	94	88	86	66	99	81	81	78
Boxes	82	123	102	88	96	101	100	73	102	87	170	108
Small	12	82	58	46	42	63	69	39	62	48	127	89
Medium	36	31	41	42	53	36	25	31	38	28	41	19
Large	34	10	3	0	1	2	6	3	2	11	2	0
Size (KB)	780	668	680	672	964	736	708	544	832	676	720	400

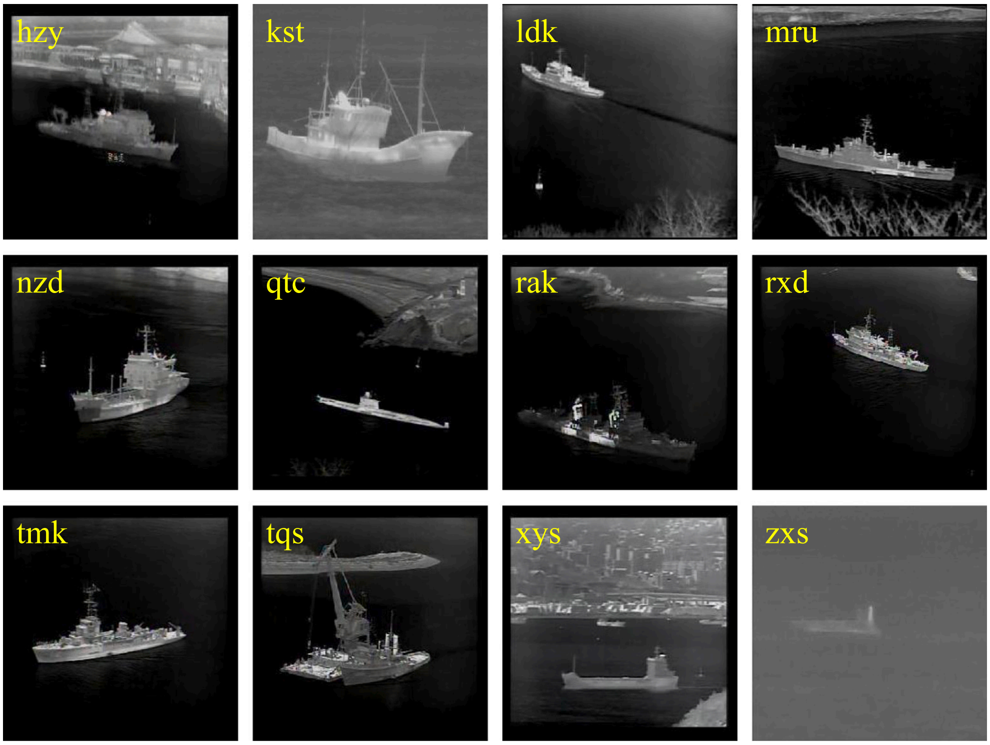

Fig. 5. The examples of the constructed Small-IRShip. Words in yellow indicate the corresponding category.

4. Experimental results and analysis

4.1. Dataset

Small-IRShip. Because there are few publicly available infrared image datasets, we construct an infrared ship dataset Small-IRShip for saliency detection. The Small-IRShip has 1002 infrared ship target images with annotations, containing 12 categories, namely hzy, kst, ldk, mru, nzd, qtc, rak, rxd, tmk, tqs, xys, zxs. Each image has a resolution of 256 × 256. Fig. 5 shows a few example images and more detailed description is shown in Table 2. In Table 2, we divide them into three sizes by the area of the target. Small are less than 32 × 32, medium are greater than 32 × 32 but less than 72 × 72, and large are greater than 72 × 72. The dataset is first created by extracting 1 image from every 60 frames of infrared ship videos. We then perform data cleaning to remove similar images. The images with missing targets are deleted as well. During the experiments, the Small-IRShip dataset is randomly divided in proportions of 8 training, 1.5 testing, and 0.5 validation.

RGB-T. RGB-T [48] is constructed for semantic segment. It contains 1569 images which has four channels (RGB-Thermal). There are eight classes in this dataset. In this paper, we first extract the last channel and ignore the other three channels. Then, we consider the eight categories above as foreground and the remaining categories as background. Training set, testing set, and validation set contain 777, 387, and 391 images, respectively. Those images without annotations are deleted.

IRSTD-1k. IRSTD-1k [49] includes 1001 infrared images. This dataset is applied in infrared small target detection. It contains different kinds of small targets in various scenes. The size of each image in this dataset is 512 × 512. For training and testing our model, IRSTD-1k is split into training set, validation set, and testing set in the ratio of 3:1:1.

Table 3
Comparison of our method with other models on the Small-IRShip dataset in terms of MAE (smaller is better), maximum F-measure (larger is better), mean F-measure (larger is better), Params (smaller is better), FLOPs (smaller is better), and FPS (faster is better). Ours* means the model with reduced channels. The data marked in red means the best results.

Methods	Year	Params(M)	FLOPs(G)	FPS	MAE	max-F_β	mean-F_β
NLDF [38]	2017	24.4	57.69	15.95	0.0074	0.9272	0.8756
PoolNet [61]	2019	49.12	120.51	7.73	0.0065	0.9189	0.8861
BASNet [39]	2019	87.06	127.56	2.48	0.0051	0.9174	0.8963
EGNet [41]	2019	108.04	156.80	4.18	0.0093	0.8803	0.8178
ITSD [93]	2020	24.86	20.82	30.35	0.0062	0.9113	0.8857
GCPANet [65]	2020	67.06	34.79	7.09	0.0063	0.9220	0.8775
F^3Net [94]	2020	25.54	8.72	40.91	0.0054	0.9146	0.8920
BPFINet [95]	2021	68.33	23.71	8.08	0.0461	0.9138	0.8355
DG-Light-NLDF [32]	2021	10.91	21.34	15.14	0.0064	0.9384	0.8725
TSERNet [75]	2022	189.64	265.92	2.04	0.0059	0.9003	0.8906
RCSBNet [96]	2022	27.25	227.12	2.19	0.0061	0.9072	0.8936
R^2Net [97]	2022	18.31	37.55	2.25	0.0056	0.9195	0.9056
DSLRDNet [98]	2023	168.60	201.92	2.82	0.0106	0.8346	0.8080
A3Net [99]	2023	16.98	42.59	5.11	0.0051	0.9231	0.9009
Ours	-	11.81	35.67	6.01	0.0040	0.9672	0.9112
Ours*	-	8.20	18.81	12.72	0.0040	0.9695	0.9063

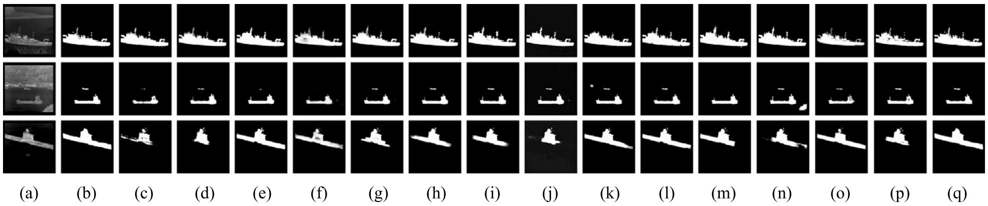

Fig. 6. Prediction results of various methods on the Small-IRShip dataset. (a) input image. (b) groundtruth. (c) NLDF. (d) PoolNet. (e) BASNet. (f) EGNet. (g) ITSD. (h) GCPANet. (i) F^3Net. (j) BPFINet. (k) TSERNet. (l) DG-Light-NLDF. (m) RCSBNet. (n) R^2Net. (o) DSLRDNet. (p) A3Net. (q) Ours.

4.2. Evaluation criteria

We use five metrics to evaluate the performance of the model, including F_β, mean absolute error (MAE), Params, FLOPs, and frames per second (FPS). F_β considers both precision $P_{precision}$ and recall R_{recall}, as given in Eq. (10).

$$F_\beta = \frac{(1+\beta^2) \times P_{precision} \times R_{recall}}{\beta^2 \times P_{precision} + R_{recall}}, \tag{10}$$

where β^2 is set to 0.3. MAE calculates the mean absolute error between predicted result P and groundtruth value Y as given in Eq. (11).

$$\mathrm{MAE} = \frac{1}{W \times H} \sum_{x,y} |\ P(x,y) - Y(x,y)\ |, \tag{11}$$

where W and H denote the width and height of the image. Params means the number of parameters to be updated. FLOPs indicates the number of floating-point operations required for the model to run once. FPS measures the speed of the model.

4.3. Implementation details

We train our network TBENet on a NVIDIA Tesla k40c GPU card using PyTorch 1.0.0 and Python 3.6. The Adam optimizer was used with an initial learning rate $lr = 1 \times 10^{-4}$. The $betas$, eps, and $weight\ decay$ are set to (0.9, 0.999), 1×10^{-8}, and 0, respectively. The $betas$ are two coefficients used for computing running averages of gradient and its square. The eps is added to the denominator to improve numerical stability. The $weight\ decay$ denotes the extent of the L_2 penalty. The overall network converges after 100 epochs with a batch size of 8. The size of input images is 352 × 352. All parameters are initialized using the $Xavier$ algorithm.

4.4. Comparison results

To verify the effectiveness of the proposed method, we carry out extensive comparative experiments on the Small-IRShip dataset with fourteen state-of-the-art saliency detection algorithms: NLDF [38], PoolNet [61], BASNet [39], EGNet [41], ITSD [93], GCPANet [65], F^3Net [94], BPFINet [95], DG-Light-NLDF [32], TSERNet [75], RCSBNet [96], R^2Net [97], DSLRDNet [98], and A3Net [99]. The hyperparameters of all methods are set to the default values in the authors' source codes. All models are retrained on the Small-IRShip dataset.

Table 4
Comparative experiments on the hzy, kst, ldk, mru, nzd, and qtc. The MAE is selected as metrics.

Methods	hzy	kst	ldk	mru	nzd	qtc
NLDF [38]	0.0139	0.0185	0.0028	0.0027	0.0065	0.0131
PoolNet [61]	0.0107	0.0165	0.0028	0.0026	0.0067	0.0127
BASNet [39]	0.0086	0.0165	0.0030	0.0020	0.0046	0.0039
EGNet [41]	0.0175	0.0236	0.0047	0.0049	0.0091	0.0080
ITSD [93]	0.0100	0.0177	0.0026	0.0022	0.0064	0.0097
GCPANet [65]	0.0116	0.0173	0.0031	0.0030	0.0074	0.0080
F^3Net [94]	0.0097	0.0187	0.0022	0.0022	0.0058	0.0069
BPFINet [95]	0.0469	0.0597	0.0393	0.0387	0.0458	0.0466
DG-Light-NLDF [32]	0.0110	0.0180	0.0031	0.0030	0.0050	0.0062
TSERNet [75]	0.0086	0.0208	0.0021	0.0017	0.0074	0.0046
RCSBNet [96]	0.0098	0.0214	0.0021	0.0019	0.0044	0.0075
R^2Net [97]	0.0096	0.0163	0.0021	0.0021	0.0045	0.0088
DSLRDNet [98]	0.0173	0.0405	0.0034	0.0026	0.0098	0.0058
A3Net [99]	0.0093	0.0161	0.0019	0.0019	0.0059	0.0077
Ours	0.0088	0.0061	0.0027	0.0028	0.0035	0.0037

Table 5
Comparative experiments on the rak, rxd, tmk, tqs, xys, and zxs. The MAE is selected as metrics.

Methods	rak	rxd	tmk	tqs	xys	zxs
NLDF [38]	0.0056	0.0029	0.0026	0.0054	0.0103	0.0020
PoolNet [61]	0.0045	0.0028	0.0027	0.0047	0.0077	0.0025
BASNet [39]	0.0037	0.0022	0.0021	0.0040	0.0083	0.0031
EGNet [41]	0.0076	0.0044	0.0041	0.0070	0.0145	0.0058
ITSD [93]	0.0052	0.0025	0.0024	0.0046	0.0082	0.0026
GCPANet [65]	0.0041	0.0031	0.0031	0.0050	0.0074	0.0021
F^3Net [94]	0.0033	0.0024	0.0023	0.0044	0.0061	0.0015
BPFINet [95]	0.0477	0.0449	0.0489	0.0511	0.0396	0.0534
DG-Light-NLDF [32]	0.0058	0.0031	0.0028	0.0052	0.0108	0.0021
TSERNet [75]	0.0051	0.0023	0.0022	0.0040	0.0101	0.0038
RCSBNet [96]	0.0063	0.0023	0.0018	0.0051	0.0092	0.0026
R^2Net [97]	0.0038	0.0023	0.0022	0.0042	0.0082	0.0022
DSLRDNet [98]	0.0090	0.0031	0.0034	0.0088	0.0204	0.0078
A3Net [99]	0.0032	0.0023	0.0021	0.0039	0.0057	0.0015
Ours	0.0025	0.0028	0.0027	0.0037	0.0045	0.0018

As shown in Table 3, our approach outperforms all previous detectors with MAE score of 0.0040, max-F_β score of 0.9672, and mean-F_β score of 0.9112. Notably, compared with the second place BASNet and A3Net, the TBENet achieves about 21.6% improvements in terms of MAE. Furthermore, the proposed model obtains the best max-F_β, which are 3.1% and 4.3% higher than DG-Light-NLDF and NLDF, respectively. Again, compared with R^2Net and A3Net, 0.6% and 1.1% increases are obtained with regards to mean-F_β. For visual comparison, we also provide some visual results, as shown in Fig. 6. From the first row, it is noted that when dealing with large targets, the TBENet is able to generate uniform saliency maps and obtain accurate edge with more details. From the second row, there are two targets in this image. NLDF cannot obtain a consistent saliency map. Some models pay more attention to the larger target so that the smaller ones are ignored. In the remaining methods, there are false detections. However, the TBENet can consider the two targets at the same time and generate clearer edges than DG-Light-NLDF. From the third row, it is obvious that many detectors cannot generate a complete saliency map when the ship is obscured. But the TBENet is able to detect the ship accurately.

In order to provide more detailed comparisons, we conduct category-based analysis experiments. From the Tables 4 and 5, we conclude that the TBENet achieves the best MAE scores on the six classes. To the specific, our algorithm obtains the MAE of 0.0061, 0.0035, 0.0037, 0.0025, 0.0037, and 0.0045 on the kst, nzd, qtc, rak, tqs, and xys, which are 62.1%, 20.5%, 5.1%, 21.9%, 5.1%, and 21.1% higher than the second best score respectively. Furthermore, our method also achieves comparable performance on the other classes. In conclusion, the TBENet has the ability to handle most scenarios.

To fully evaluate the performance of our algorithm, we conduct the complexity comparison experiment. The Params, FLOPs, and FPS are selected as metrics. From the third and fifth columns in Table 3, the Params of TBENet is 11.81M, which is the second lowest value and is slightly higher than the DG-Light-NLDF. The lightweight of the model is enhanced when we reduce the number of channels, obtaining the smallest Params as well as the second smallest FLOPs. In conclusion, the TBENet achieves a balance between accuracy and efficiency.

4.5. Ablation experiments

4.5.1. Influence of different components on model performance

To validate the contribution of each component of our model to the TBENet, we perform detailed ablation experiments. In this section, we verify the effectiveness of the SAMM, the HLLB module, the MCIM, the DMIM, the average pooling layer, and the

Table 6
Ablation study on the Small-IRShip dataset.

Methods	MAE	max-F_β	mean-F_β
w/o SAMM	0.0042	0.9644	0.8898
w/o HLLB	0.0053	0.9551	0.8849
w/o MCIM	0.0051	0.9580	0.8606
w/o DMIM	0.0056	0.9597	0.8464
w/o AvgPool	0.0041	0.9645	0.9018
w/o Edge	0.0054	0.9542	0.8791
TBENet	0.0040	0.9672	0.9112

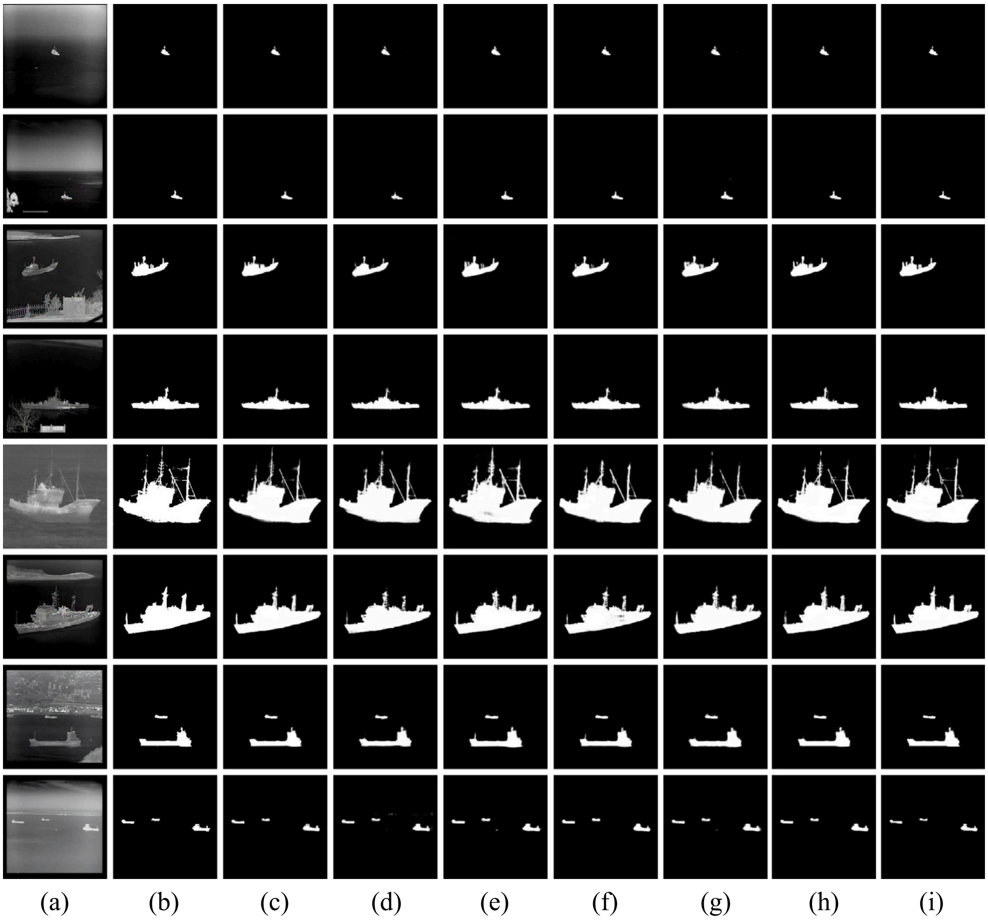

Fig. 7. Qualitative results comparison of ablation study. (a) input image. (b) groundtruth. (c) w/o AvgPool. (d) w/o Edge. (e) w/o DMIM. (f) w/o HLLB. (g) w/o MCIM. (h) w/o SAMM. (i) TBENet.

two-branch structure. Table 6 gives the results of the ablation experiments. In Table 6, w/o SAMM denotes that the spatial attention module is replaced by a convolution layer. w/o HLLB indicates that we use standard convolution rather than the high-level linear bottleneck module. w/o MCIM denotes that three feature maps containing different clues are integrated via linear interpolation and concatenation instead of using MCIM. w/o DMIM means that the same operation as the w/o MCIM is used. w/o Edge indicates that the two-branch structure is absent and the saliency map is predicted directly. w/o AvgPool means that the average pooling is replaced by the max pooling.

Specifically, from the first row, the performance is decreased in terms of three metrics when the SAMM is not used, representing that the SAMM can enhance the mid-level feature maps. From the second row, the MAE degrades by 32.5%, the max-F_β decreases by 1.3%, and the mean-F_β declines by 2.9%, which demonstrates that the HLLB can effectively improve the performance of the model by providing a larger receptive field. From the third row, the MAE is reduced by 27.5%, the max-F_β is decreased by 1.0%, and the mean-F_β is decreased by 5.6%. It can be seen that the MCIM can effectively leverage the various clues with high efficiency. From the fourth row, by imposing the depthwise multi-scale integration module on the TBENet, the MAE, the max-F_β, and the mean-F_β are improved. In particular, the MAE, the max-F_β, and the mean-F_β have a 40.0%, 0.8%, and 7.1% improvement, respectively. It

Table 7
Influence of HLLB with different kernel sizes on model performance.

Methods	Params(M)	FLOPs(G)	FPS	MAE	max-F_β	mean-F_β
w/o HLLB	19.74	46.69	5.20	0.0053	0.9551	0.8849
HLLB-1,1-3,3-5,5	11.64	35.31	6.10	0.0043	0.9650	0.9094
HLLB-5,5-7,7-9,9	12.08	36.21	5.81	0.0041	0.9663	0.8942
HLLB-3,3-5,5-7,7	11.81	35.67	6.01	0.0040	0.9672	0.9112

Table 8
Experimental results obtained by models with different loss functions.

Loss	MAE	max-F_β	mean-F_β
$L_b + L_c$	0.0063	0.9273	0.8302
$L_b^* + L_c + L_s$	0.0050	0.9583	0.9033
$L_b + L_c + L_s$	0.0040	0.9672	0.9112

Table 9
Experimental results predicted by models trained with datasets divided by different ratios.

Ratios	MAE	max-F_β	mean-F_β
6:2:2	0.0041	0.9630	0.9080
7:2:1	0.0042	0.9738	0.9109
8:1.5:0.5 (Ours)	0.0040	0.9672	0.9112

can be seen that the DMIM has the ability to better integrate features at different scales. Furthermore, the max pooling layer is often used to reduce the resolution in VGG16. However, there may be information loss. Therefore, we select the average pooling in this paper. From the fifth row, average pooling has little impact on performance. Furthermore, when the contour prediction branch is overlooked, the performance of the model shows different degrees of degradation. It is noted that guided by the edge information, the TBENet is able to generate accurate saliency maps. In conclusion, the quantitative results obtained from the ablation experiments demonstrate that each component is indispensable.

For visual representation, some visualization results are shown in Fig. 7. Fig. 7 shows the detection results of different methods when facing the small target (first and second rows), medium target (third and fourth rows), large target (fifth and sixth rows), and multi-target (seventh and eighth rows) situations. Specifically, the rods w/o AvgPool detection are much thinner than the groundtruth. The w/o Edge detects the edges of defects. Blurred detail information is detected by the w/o DMIM. Incomplete detail information appears in detection results of the w/o HLLB. However, the TBENet achieves more accurate detection results.

4.5.2. Influence of HLLB with different kernel sizes on model performance

Influence of HLLB with different kernel sizes on predicted results will be exhibited in this section. Table 7 reveals experimental results. HLLB-$k1$, $k1$-$k2$, $k2$-$k3$, $k3$ means that the size of the convolution kernels chosen for the three deepwise convolutions are $k1$, $k2$, and $k3$ respectively. First, compared with the method using standard convolution (first row), these models using HLLB achieve improvements in all three evaluation metrics. Second, when the kernel size is 3, 5, and 7, the Params, the FLOPs, and the FPS of the model are slightly lower than the optimal values, however, the model obtains the best results with a MAE of 0.0040, a max-F_β of 0.9672, and a mean-F_β of 0.9112. It shows that the HLLB improves model performance while reducing model complexity.

4.5.3. Influence of different loss functions on model performance

The effectiveness of the loss function used in this paper will be verified in this section. As shown in Table 8, L_b, L_c, and L_s denote the saliency prediction loss, the contour prediction loss, and the SSIM loss, respectively (see Section 3.6). L_b^* means the saliency prediction loss without edge weighting. It is noted that when L_s is not used, the MAE, the max-F_β, and the mean-F_β degrade by 57.5%, 4.1%, and 8.9%, respectively. Furthermore, compared with the model without considering edge weights, our algorithm obtains 20%, 0.9%, and 0.9% increases in terms of MAE, max-F_β, and mean-F_β. In conclusion, both L_b and L_s boost the model's performance to varying degrees.

4.5.4. Influence of the proportion of training, validation and testing sets on model performance

In this section, in order to choose the best way to divide the dataset, we choose three different ratios to divide the dataset. From the Table 9, as the percentage of the training set gradually increases, the model tends to achieve superior performance, which is consistent with general knowledge. When the Small-IRShip dataset is split into training set, validation set, and testing set in the ratio of 0.8:0.15:0.05, our model achieves the best MAE and mean-F_β, confirming the justification of such ratios.

4.5.5. Influence of different pooling techniques in SAMM on model performance

In SAMM, max pooling highlights the target features, and average pooling captures the detailed information of the local area and facilitates the calculation of the dependency between pixels. In this section, we verify their effect by using max pooling and

Table 10
Experimental results predicted by models with different SAMMs.

Methods	MAE	max-F_β	mean-F_β
SAMM_mean	0.0041	0.9656	0.8974
SAMM_max	0.0041	0.9657	0.9067
Ours	0.0040	0.9672	0.9112

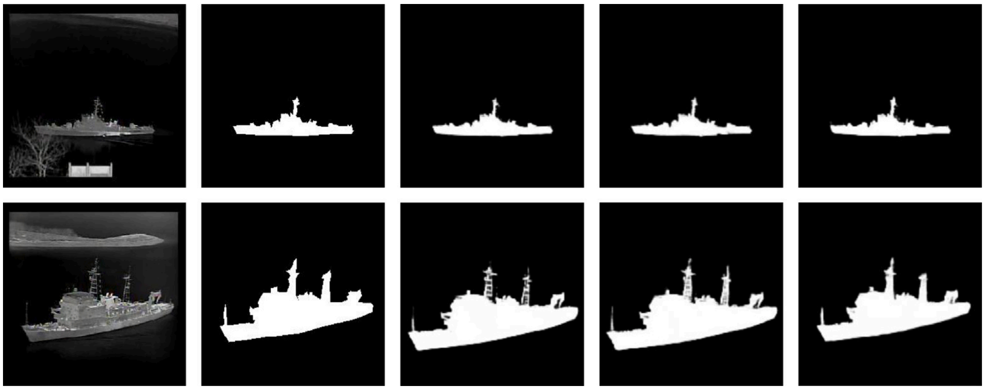

Fig. 8. Visual results comparison of different SAMMs.

Table 11
Experimental results predicted by models with different fusion strategies.

Methods	MAE	max-F_β	mean-F_β
MCIM_hlm	0.0043	0.9651	0.8931
MCIM_lmh	0.0041	0.9656	0.8969
Ours	0.0040	0.9672	0.9112

average pooling, respectively. As shown in Table 10, suboptimal results are obtained for the model by using only max pooling or average pooling. And only when both of the pooling are used, the model obtains optimal results. From Fig. 8, it is noticed that max pooling can generates distinct contour features and mean pooling is able to maintain clear detail information. However, our model contains the two pooling techniques so that accurate saliency map is obtained.

4.5.6. Influence of different fusion strategies on model performance

In MCIM, we first fuse the high-level features with the mid-level features and the obtained result is then fused with the low-level features. In order to verify the impact of different fusion strategies on the model performance, we chose three fusion methods. As shown is Table 11, MCIM_hlm denotes that the high-level features is first fused with the low-level features and the resulting is then fused with the mid-level features. MCIM_lmh denotes that the low-level features is first fused with the mid-level features and the resulting is then fused with the high-level features. From Table 11 and Fig. 9, we can conclude that our proposed fusion method achieves optimal performance. The underlying reason is that both MCIM_hlm and MCIM_lmh require high-level features to be upsampled to the same size as the low-level feature maps, and this large transformation leads to loss of spatial information and produces inaccurate saliency maps.

4.6. Generalizability analysis

To validate the generalization ability of the model, we carry out comparison and ablation experiments on the public benchmark RGB-T [48] and IRSTD-1k [49]. In this section, all models are trained from scratch.

RGB-T. From Table 12, the TBENet achieves the best max-F_β and mean-F_β, the third MAE. Typically, the proposed model obtains an MAE score of 0.0321, which is lower than GCPANet and F^{3}Net. However, our method achieves the best max-F_β, which is 2.0%, 2.5%, and 2.6% higher than BPFINet, F^{3}Net, and GCPANet, respectively. Furthermore, our approach also obtains leading performance in terms of mean-F_β. When the number of channels of feature maps are decreased, mean-F_β and max-F_β degrade slightly, but the MAE increases by 0.3%. In Fig. 10, we offer qualitative comparisons. As we can see from the first row, some detectors fail to generate accurate saliency maps, including NLDF, PoolNet, EGNet, and DG-Light-NLDF and other methods detect a different object, such as F^{3}Net, BPFINet, and TSERNet. As a comparison, the TBENet overlooks these non-targets and obtains an accurate saliency map. From the second row, when there are multiple targets with different categories, PoolNet and EGNet

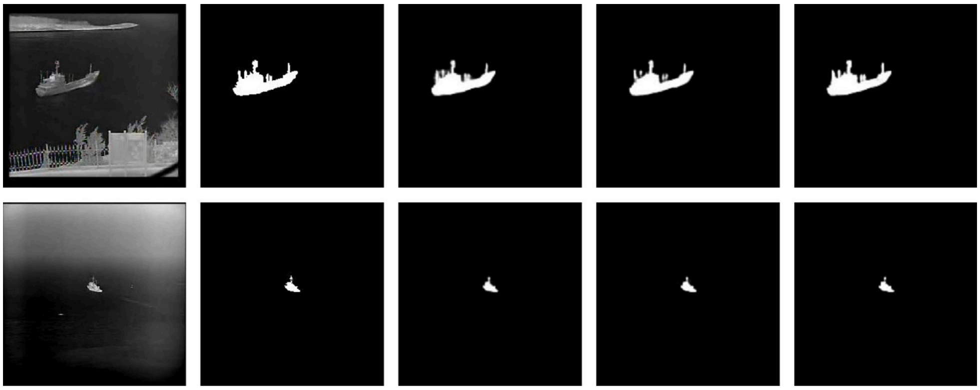

Fig. 9. Visual results comparison of different fusion strategies.

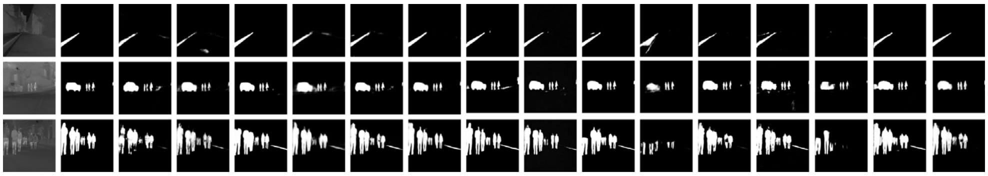

Fig. 10. Visual results of various methods on the RGB-T dataset. (a) input image. (b) groundtruth. (c) NLDF. (d) PoolNet. (e) BASNet. (f) EGNet. (g) ITSD. (h) GCPANet. (i) F^3Net. (j) BPFINet. (k) TSERNet. (l) DG-Light-NLDF. (m) RCSBNet. (n) R^2Net. (o) DSLRDNet. (p) A3Net. (q) Ours.

cannot extract the car from the background because of the insignificant differences between the foreground and background. DG-Light-NLDF fails to capture the car accurately. Some non-targeted areas are detected by the NLDF, BASNet, BPFINet, and other methods. However, the TBENet captures both car and humans while not being affected by interfering objects. From the third row, when multiple targets of the same category emerge, NLDF and DG-Light-NLDF cannot generate uniform saliency maps while other detectors focus on erroneous targets. However, the TBENet detects all targets and produces a near-accurate saliency map. In brief, our technique achieves competitive performance on the RGB-T dataset.

IRSTD-1k. The IRSTD-1k dataset is chosen to validate the effectiveness of our model against small targets. As we can see from Table 12, the TBENet achieves the best MAE, the second max-F_β, and the fourth mean-F_β. In more detail, our model is tied for the first place with other state-of-the-art algorithms with an MAE score of 0.0002. And the max-F_β of our method is 0.7688, which is lower than the PoolNet and is about 4.7% higher than the EGNet. In addition, our approach obtains a mean-F_β of 0.6318, outperforming most algorithms. When we reduce the number of channels of the feature maps output by each stage in the decoder, the model efficiency improves substantially, although the model's performance decreases slightly. Furthermore, we also present some specific examples in Fig. 11. The results from the first and second rows indicate that the missed detection occurs in the prediction of BASNet and TSERNet. The unclear edges are generated by EGNet, ITSD, and GCPANet. However, our technique is able to detect all targets with clear edges. And then, we can infer according to the third and fourth rows that BASNet and ITSD are affected by distractors. Most algorithms detect the target whose shape is inconsistent with the groundtruth. In contrast, our model generates accurate contours. From the fifth and sixth rows, most methods fail to capture the target and the DG-Light-NLDF detects blurred edges. But the TBENet is able to compute probability distribution accurately.

In order to highlight the contribution of each component to the model, we perform ablation experiments on the RGB-T and IRSTD-1k. As shown in Table 13, when the model lacks any one of the components, its performance produces a varying degree of degradation. Notably, the MAE, max-F_β, and mean-F_β decrease by 26.2%, 7.7%, and 10.9% on the RGB-T, respectively, when the contour predication branch is ignored. The w/o DMIM achieves the MAE of 0.0405, the max-F_β of 0.6968, and the mean-F_β of 0.6525 on the RGB-T, which are lower than the TBENet. Therefore, the DMIM and the contour predication branch are beneficial for refining feature maps and generating accurate contour. What is more, the average pooling is critical to our model when the IRSTD-1k is selected. Specifically, max-F_β, and mean-F_β decline by 32.1% and 30.7%. The potential reason is that small targets are filtered out by max pooling. In conclusion, the effectiveness of each component is validated on different datasets.

5. Conclusion

In this paper, we propose a novel pipeline for saliency detection in infrared images, named TBENet. We employ a two-branch structure consisting of a contour prediction branch and a saliency map generation branch. The contour prediction branch is inserted

Table 12
Comparison of our method with other methods on the RGB-T and IRSTD-1k datasets. The data marked in red, green, and blue means the first, second, and third best results, respectively.

Methods	RGB-T			IRSTD-1k		
	MAE	max-F_β	mean-F_β	MAE	max-F_β	mean-F_β
NLDF [38]	0.0458	0.7249	0.6361	0.0002	0.6303	0.5326
PoolNet [61]	0.0406	0.6954	0.6620	0.0002	0.7871	0.7104
BASNet [39]	0.0338	0.7261	0.7030	0.0002	0.6801	0.6138
EGNet [41]	0.0517	0.6545	0.5983	0.0002	0.7343	0.6249
ITSD [93]	0.0351	0.7343	0.6880	0.0002	0.6620	0.6030
GCPANet [65]	0.0298	0.7616	0.7214	0.0002	0.6675	0.5463
F^3Net [94]	0.0283	0.7624	0.7268	0.0002	0.7070	0.6447
BPFINet [95]	0.0427	0.7657	0.7269	0.1780	0.6994	0.4178
DG-Light-NLDF [32]	0.0551	0.6375	0.5939	0.0002	0.6825	0.5827
TSERNet [75]	0.0420	0.6812	0.6679	0.0002	0.7127	0.6464
RCSBNet [96]	0.0359	0.7189	0.6940	0.0002	0.6362	0.6269
R^2Net [97]	0.0361	0.7380	0.6820	0.0002	0.7113	0.6298
DSLRDNet [98]	0.0606	0.4799	0.4446	0.0002	0.6519	0.6012
A3Net [99]	0.0331	0.7353	0.7011	0.0002	0.6897	0.6268
Ours	0.0321	0.7811	0.7310	0.0002	0.7688	0.6318
Ours*	0.0320	0.7637	0.7205	0.0002	0.7593	0.6317

Table 13
Ablation experiments on the RGB-T and IRSTD-1k datasets.

Methods	RGB-T			IRSTD-1k		
	MAE	max-F_β	mean-F_β	MAE	max-F_β	mean-F_β
w/o SAMM	0.0330	0.7472	0.7128	0.0002	0.7604	0.6246
w/o HLLB	0.0401	0.7221	0.6629	0.0002	0.6329	0.5139
w/o MCIM	0.0403	0.7070	0.6390	0.0003	0.6121	0.4865
w/o DMIM	0.0405	0.6968	0.6525	0.0002	0.6897	0.5564
w/o AvgPool	0.0335	0.7545	0.7097	0.0002	0.5218	0.4380
w/o edge	0.0405	0.7207	0.6514	0.0002	0.7430	0.6066
Ours	0.0321	0.7811	0.7310	0.0002	0.7688	0.6318

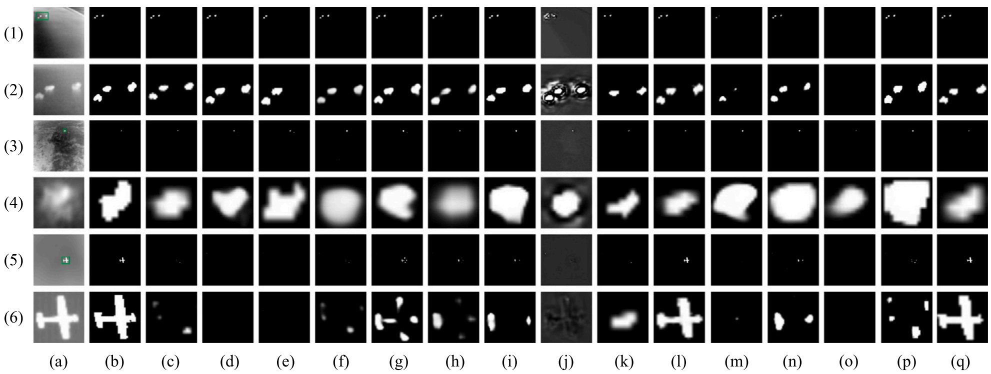

Fig. 11. Qualitative results of various methods on the IRSTD-1k dataset. (a) input image. (b) groundtruth. (c) NLDF. (d) PoolNet. (e) BASNet. (f) EGNet. (g) ITSD. (h) GCPANet. (i) F^3Net. (j) BPFINet. (k) TSERNet. (l) DG-Light-NLDF. (m) RCSBNet. (n) R^2Net. (o) DSLRDNet. (p) A3Net. (q) Ours. The targets are denoted by a green bounding box in the first, third, and fifth rows. And the second, fourth, and sixth rows present the magnified objects.

on top of the U-shape structure to extract the contour information of the target. Guided by the obtained contour information, the saliency map generation branch can generate unified saliency map by using an edge-weighted loss to supervise the entire training process. Moreover, we introduce a depthwise multi-scale integration module for fusing deep and shallow features and a multi-clue integration module for aggregating features from high, middle, and low levels. Furthermore, we effectively reduce the number of parameters by incorporating a lightweight high-level linear bottleneck module and reducing the number of channels of the feature maps from each stage of the decoder. Comparison results with ten state-of-the-art models demonstrate that our proposed method achieves leading accuracy with high efficiency. Since our model requires manual annotation of the edges of an image, it takes a lot of time. Therefore, in future research, we will try to design an edge-aware attention that allows the network to adaptively focus on the edges of objects without extra supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgment

This work is supported by General Project of Science and Technology Plan of Beijing Municipal Education Commission (KM202110005028), National Natural Science Foundation of China (62176009, 61806013, 61906005), International Research Cooperation Seed Fund of Beijing University of Technology (2021A01), and Project of Interdisciplinary Research Institute of Beijing University of Technology (2021020101).

References

[1] Zhang R, Li L, Zhang Q, Zhang J, Xu L, Zhang B, Wang B. Differential feature awareness network within antagonistic learning for infrared-visible object detection. IEEE Trans Circuits Syst Video Technol 2023.
[2] Zhang T, Jiang G, Liu Z, ur Rehman S, Li Y. Advanced integrated segmentation approach for semi-supervised infrared ship target identification. Alexandria Eng J 2024;87:17–30.
[3] Zhang T, Shen H, ur Rehman S, Liu Z, Li Y, ur Rehman O. Two-stage domain adaptation for infrared ship target segmentation. IEEE Trans Geosci Remote Sens 2023.
[4] Tan H, Ye T, ur Rehman S, ur Rehman O, Tu S, Ahmad J. A novel routing optimization strategy based on reinforcement learning in perception layer networks. Comput Netw 2023;237:110105.
[5] Xu G, Liang Y, Tu S, ur Rehman S. A spatial-temporal integration analysis to classify dynamic functional connectivity for brain disease diagnosis. In: International conference on adaptive and intelligent systems. Springer; 2022, p. 549–58.
[6] Tu S, Li W, Ai X, Li H, Yue Q, Rehman SU. A hybrid deep learning model for breast cancer detection and classification. In: Proceedings of the 2023 13th international conference on communication and network security. 2023, p. 350–3.
[7] Rehman SU, Tu S, Huang Y, Rehman OU. A benchmark dataset and learning high-level semantic embeddings of multimedia for cross-media retrieval. IEEE Access 2018;6:67176–88.
[8] Rehman SU, Tu S, Huang Y, Yang Z. Face recognition: A novel un-supervised convolutional neural network method. In: 2016 IEEE international conference of online analysis and computing science. ICOACS, IEEE; 2016, p. 139–44.
[9] Qureshi M, Arbab MA, et al. Deep learning-based forecasting of electricity consumption. Sci Rep 2024;14(1):1–11.
[10] Li M, Tu S, Rehman Su. Facial expression recognition from occluded images using deep convolution neural network with vision transformer. In: International conference on image and graphics. Springer; 2023, p. 289–99.
[11] ur Rehman S, Huang Y, Tu S, Ahmad B. Learning a semantic space for modeling images, tags and feelings in cross-media search. In: Trends and applications in knowledge discovery and data mining: PAKDD 2019 workshops, BDM, DLKT, LDRC, PAISI, WeL, Macau, China, April 14–17, 2019, revised selected papers 23. Springer; 2019, p. 65–76.
[12] Li X, Zhang T, Liu Z, Liu B, ur Rehman S, Rehman B, Sun C. Saliency guided siamese attention network for infrared ship target tracking. IEEE Trans Intell Veh 2024.
[13] Li J, Han L, Zhang C, Li Q, Liu Z. Spherical convolution empowered viewport prediction in 360 video multicast with limited FoV feedback. ACM Trans Multimedia Comput Commun Appl 2023;19(1):1–23.
[14] Li J, Zhang C, Liu Z, Hong R, Hu H. Optimal volumetric video streaming with hybrid saliency based tiling. IEEE Trans Multimed 2022.
[15] Liu K, Jiang Z, Lalancette RA, Tang X, Jakle F. Near-infrared-absorbing B–N lewis pair-functionalized anthracenes: electronic structure tuning, conformational isomerism, and applications in photothermal cancer therapy. J Am Chem Soc 2022;144(41):18908–17.
[16] Zheng W, Lu S, Yang Y, Yin Z, Yin L. Lightweight transformer image feature extraction network. PeerJ Comput Sci 2024;10:e1755.
[17] Mi C, Liu Y, Zhang Y, Wang J, Feng Y, Zhang Z. A vision-based displacement measurement system for foundation pit. IEEE Trans Instrum Meas 2023.
[18] Sun G, Xu Z, Yu H, Chen X, Chang V, Vasilakos AV. Low-latency and resource-efficient service function chaining orchestration in network function virtualization. IEEE Internet Things J 2019;7(7):5760–72.
[19] Sun G, Zhu G, Liao D, Yu H, Du X, Guizani M. Cost-efficient service function chain orchestration for low-latency applications in NFV networks. IEEE Syst J 2018;13(4):3877–88.
[20] Ding P, Zhang Y, Jia P, Chang X-L, Liu R. Ship detection on sea surface based on visual saliency. Tien Tzu Hsueh Pao/Acta Electron Sin 2018;46(1):127–34. http://dx.doi.org/10.3969/j.issn.0372-2112.2018.01.018.
[21] Dong Y, Xu B, Liao T, Yin C, Tan Z. Application of local-feature-based 3D point cloud stitching method of low-overlap point cloud to aero-engine blade measurement. IEEE Trans Instrum Meas 2023.
[22] Qi F, Tan X, Zhang Z, Chen M, Xie Y, Ma L. Glass makes blurs: Learning the visual blurriness for glass surface detection. IEEE Trans Ind Inf 2024.
[23] Fu C, Yuan H, Xu H, Zhang H, Shen L. TMSO-Net: Texture adaptive multi-scale observation for light field image depth estimation. J Vis Commun Image Represent 2023;90:103731.
[24] Cheng B, Wang M, Zhao S, Zhai Z, Zhu D, Chen J. Situation-aware dynamic service coordination in an IoT environment. IEEE/ACM Trans Netw 2017;25(4):2082–95.
[25] Dai M, Sun G, Yu H, Niyato D. Maximize the long-term average revenue of network slice provider via admission control among heterogeneous slices. IEEE/ACM Trans Netw 2023.
[26] Di Y, Li R, Tian H, Guo J, Shi B, Wang Z, Yan K, Liu Y. A maneuvering target tracking based on fastIMM-extended viterbi algorithm. Neural Comput Appl 2023;1–10.
[27] Zhu W, Liang S, Wei Y, Sun J. Saliency optimization from robust background detection. In: 2014 IEEE conference on computer vision and pattern recognition. 2014, p. 2814–21. http://dx.doi.org/10.1109/CVPR.2014.360.
[28] Liu T, Sun J, Zheng N-N, Tang X, Shum H-Y. Learning to detect a salient object. In: 2007 IEEE conference on computer vision and pattern recognition. 2007, p. 1–8. http://dx.doi.org/10.1109/CVPR.2007.383047.

[29] Yang C, Zhang L, Lu H, Ruan X, Yang M-H. Saliency detection via graph-based manifold ranking. In: 2013 IEEE conference on computer vision and pattern recognition. 2013, p. 3166–73. http://dx.doi.org/10.1109/CVPR.2013.407.
[30] Hou Q, Cheng M-M, Hu X, Borji A, Tu Z, Torr P. Deeply supervised salient object detection with short connections. In: IEEE conference on computer vision and pattern recognition. CVPR, 2017, p. 5300–9. http://dx.doi.org/10.1109/CVPR.2017.563.
[31] Zhang L, Dai J, Lu H, He Y, Wang G. A bi-directional message passing model for salient object detection. In: 2018 IEEE conference on computer vision and pattern recognition. 2018, p. 1741–50. http://dx.doi.org/10.1109/CVPR.2018.00187.
[32] Liu Z, Zhang X, Jiang T, Zhang T, Liu B, Waqas M, Li Y. Infrared salient object detection based on global guided lightweight non-local deep features. Infrared Phys Technol 2021;115:103672. http://dx.doi.org/10.1016/j.infrared.2021.103672.
[33] Shi Y, Xi J, Hu D, Cai Z, Xu K. RayMVSNet++: learning ray-based 1D implicit fields for accurate multi-view stereo. IEEE Trans Pattern Anal Mach Intell 2023.
[34] Han X, Zhao C, Wang S, Pan Z, Jiang Z, Tang X. Multifunctional TiO2/C nanosheets derived from 3D metal–organic frameworks for mild-temperature-photothermal-sonodynamic-chemodynamic therapy under photoacoustic image guidance. J Colloid Interface Sci 2022;621:360–73.
[35] Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H. MobileNets: Efficient convolutional neural networks for mobile vision applications. 2017, arXiv preprint arXiv:1704.04861.
[36] Han K, Wang Y, Tian Q, Guo J, Xu C, Xu C. GhostNet: More features from cheap operations. In: IEEE/CVF conference on computer vision and pattern recognition. CVPR, 2020, p. 1577–86. http://dx.doi.org/10.1109/CVPR42600.2020.00165.
[37] Li Y, Chen Y, Dai X, Chen D, Liu M, Yuan L, Liu Z, Zhang L, Vasconcelos N. MicroNet: Towards image recognition with extremely low FLOPs. 2020, arXiv preprint arXiv:2011.12289.
[38] Luo Z, Mishra A, Achkar A, Eichel J, Li S, Jodoin P-M. Non-local deep features for salient object detection. In: IEEE conference on computer vision and pattern recognition. CVPR, 2017, p. 6593–601. http://dx.doi.org/10.1109/CVPR.2017.698.
[39] 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, p. 7471–81. http://dx.doi.org/10.1109/CVPR.2019.00766.
[40] Tu Z, Ma Y, Li C, Tang J, Luo B. Edge-guided non-local fully convolutional network for salient object detection. IEEE Trans Circuits Syst Video Technol 2021;31(2):582–93. http://dx.doi.org/10.1109/TCSVT.2020.2980853.
[41] Zhao J, Liu J-J, Fan D-P, Cao Y, Yang J, Cheng M-M. EGNet: Edge guidance network for salient object detection. In: 2019 IEEE/CVF international conference on computer vision. ICCV, 2019, p. 8778–87. http://dx.doi.org/10.1109/ICCV.2019.00887.
[42] Guo T, Yuan H, Wang L, Wang T. Rate-distortion optimized quantization for geometry-based point cloud compression. J Electron Imaging 2023;32(1). 013047–013047.
[43] Xing J, Yuan H, Hamzaoui R, Liu H, Hou J. GQE-Net: a graph-based quality enhancement network for point cloud color attribute. IEEE Trans Image Process 2023;32:6303–17.
[44] Wang D, Zhang W, Wu W, Guo X. Soft-label for multi-domain fake news detection. IEEE Access 2023.
[45] Ma D, Fang H, Wang N, Lu H, Matthews J, Zhang C. Transformer-optimized generation, detection, and tracking network for images with drainage pipeline defects. Comput-Aided Civ Infrastruct Eng 2023;38(15):2109–27.
[46] Lei Y, Yanrong C, Hai T, Ren G, Wenhuan W. DGNet: An adaptive lightweight defect detection model for new energy vehicle battery current collector. IEEE Sens J 2023.
[47] Ma S, Chen Y, Yang S, Liu S, Tang L, Li B, Li Y. The autonomous pipeline navigation of a cockroach bio-robot with enhanced walking stimuli. Cyborg Bionic Syst 2023;4:0067.
[48] Ha Q, Watanabe K, Karasawa T, Ushiku Y, Harada T. MFNet: Towards real-time semantic segmentation for autonomous vehicles with multi-spectral scenes. In: 2017 IEEE/RSJ international conference on intelligent robots and systems. IROS, 2017, p. 5108–15. http://dx.doi.org/10.1109/IROS.2017.8206396.
[49] Zhang M, Zhang R, Yang Y, Bai H, Zhang J, Guo J. ISNet: Shape matters for infrared small target detection. In: 2022 IEEE/CVF conference on computer vision and pattern recognition. CVPR, 2022, p. 867–76. http://dx.doi.org/10.1109/CVPR52688.2022.00095.
[50] Rehman SU, Tu S, Rehman OU, Huang Y, Magurawalage CMS, Chang C-C. Optimization of CNN through novel training strategy for visual classification problems. Entropy 2018;20(4):290.
[51] Zhang T, Waqas M, Liu Z, Tu S, Halim Z, Rehman SU, Li Y, Han Z. A fusing framework of shortcut convolutional neural networks. Inform Sci 2021;579:685–99.
[52] ur Rehman S, Tu S, Waqas M, Huang Y, ur Rehman O, Ahmad B, Ahmad S. Unsupervised pre-trained filter learning approach for efficient convolution neural network. Neurocomputing 2019;365:171–90.
[53] Ullah A, Rehman Su, Tu S, Mehmood RM, Fawad, Ehatisham-ul Haq M. A hybrid deep CNN model for abnormal arrhythmia detection based on cardiac ECG signal. Sensors 2021;21(3):951.
[54] Uesugi K, Mayama H, Morishima K. Analysis of rowing force of the water strider middle leg by direct measurement using a bio-appropriating probe and by indirect measurement using image analysis. Cyborg Bionic Syst 2023;4:0061.
[55] Ma J, Hu J. Safe consensus control of cooperative-competitive multi-agent systems via differential privacy. Kybernetika 2022;58(3):426–39.
[56] Zhao L, Qu S, Xu H, Wei Z, Zhang C. Energy-efficient trajectory design for secure SWIPT systems assisted by UAV-IRS. Veh Commun 2024;45:100725.
[57] Ge Y, Zhang Q, Xiang T-Z, Zhang C, Zhang J, Bi H. GSNNet: Group semantic-guided neighbor interaction network for co-salient object detection. Comput Vis Image Underst 2023;227:103611. http://dx.doi.org/10.1016/j.cviu.2022.103611.
[58] Ren G, Xie Y, Dai T, Stathaki T. Progressive multi-scale fusion network for rgb-d salient object detection. Comput Vis Image Underst 2022;223:103529. http://dx.doi.org/10.1016/j.cviu.2022.103529.
[59] Huang M, Liu Z, Ye L, Zhou X, Wang Y. Saliency detection via multi-level integration and multi-scale fusion neural networks. Neurocomputing 2019;364:310–21. http://dx.doi.org/10.1016/j.neucom.2019.07.054.
[60] Luo H, Han G, Wu X, Liu P, Yang H, Zhang X. Cascaded hourglass feature fusing network for saliency detection. Neurocomputing 2021;428:206–17. http://dx.doi.org/10.1016/j.neucom.2020.11.058.
[61] Liu J-J, Hou Q, Cheng M-M, Feng J, Jiang J. A simple pooling-based design for real-time salient object detection. In: IEEE/CVF conference on computer vision and pattern recognition. CVPR, 2019, p. 3912–21. http://dx.doi.org/10.1109/CVPR.2019.00404.
[62] Zhao Z, Xia C, Xie C, Li J. Complementary trilateral decoder for fast and accurate salient object detection. In: Proceedings of the 29th ACM international conference on multimedia. 2021, p. 4967–75.
[63] Song X, Guo F, Zhang L, Lu X, Hei X. Salient object detection with dual-branch stepwise feature fusion and edge refinement. IEEE Trans Circuits Syst Video Technol 2023. http://dx.doi.org/10.1109/TCSVT.2023.3312859, 1–1.
[64] Yan L, Geng G, Zhang Q, Feng L, Liu Y, Ge X, Jia H. Multiscale feature aggregation network for salient object detection in optical remote sensing images. IEEE Sens J 2023;23(16):18362–73. http://dx.doi.org/10.1109/JSEN.2023.3286373.
[65] Chen Z, Xu Q, Cong R, Huang Q. Global context-aware progressive aggregation network for salient object detection. In: Proceedings of the AAAI conference on artificial intelligence. Vol. 34, 2020, p. 10599–606.
[66] Chen L, Liu H, Mo J, Zhang D, Yang J, Lin F, Zheng Z, Jia R. Cross channel aggregation similarity network for salient object detection. Int J Mach Learn Cybern 2022;13:2153–69. http://dx.doi.org/10.1007/s13042-022-01512-y.

[67] Li C, Xuan S, Liu F, Chang E, Wu H. Global attention network for collaborative saliency detection. Int J Mach Learn Cybern 2023;14:407–17. http://dx.doi.org/10.1007/s13042-022-01531-9.

[68] Sun F, Zhang K, Yuan X, Zhao C. Feature enhancement and fusion for RGB-t salient object detection. In: 2023 IEEE international conference on image processing. ICIP, 2023, p. 1300–4. http://dx.doi.org/10.1109/ICIP49359.2023.10222404.

[69] Huang R, Zhao Q, Xing Y, Gao S, Xu W, Zhang Y, Fan W. A saliency enhanced feature fusion based multiscale RGB-D salient object detection network. In: 2024 IEEE international conference on acoustics, speech and signal processing. ICASSP, 2024, p. 9356–60. http://dx.doi.org/10.1109/ICASSP48485.2024.10447807.

[70] Zhao L, Xu H, Qu S, Wei Z, Liu Y. Joint trajectory and communication design for UAV-assisted symbiotic radio networks. IEEE Trans Veh Technol 2024.

[71] Liu Y, Zhao B, Zhao Z, Liu J, Lin X, Wu Q, Susilo W. SS-DID: A secure and scalable Web3 decentralized identity utilizing multi-layer sharding blockchain. IEEE Internet Things J 2024.

[72] Ding Y, Zhang W, Zhou X, Liao Q, Luo Q, Ni LM. FraudTrip: Taxi fraudulent trip detection from corresponding trajectories. IEEE Internet Things J 2020;8(16):12505–17.

[73] Chen X, Zhang Q, Zhang L. Edge-aware salient object detection network via context guidance. Image Vis Comput 2021;110:104166. http://dx.doi.org/10.1016/j.imavis.2021.104166.

[74] Wu Z, Su L, Huang Q. Stacked cross refinement network for edge-aware salient object detection. In: 2019 IEEE/CVF international conference on computer vision. ICCV, 2019, p. 7263–72. http://dx.doi.org/10.1109/ICCV.2019.00736.

[75] 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. http://dx.doi.org/10.1109/TIM.2022.3200114.

[76] Zheng Q, Zheng L, Bai Y, Liu H, Deng J, Li Y. Boundary-aware network with two-stage partial decoders for salient object detection in remote sensing images. IEEE Trans Geosci Remote Sens 2023;61:1–13. http://dx.doi.org/10.1109/TGRS.2023.3260825.

[77] Li J, Wang Z, Pan Z, Liu Q, Guo D. Looking at boundary: Siamese densely cooperative fusion for salient object detection. IEEE Trans Neural Netw Learn Syst 2023;34(7):3580–93. http://dx.doi.org/10.1109/TNNLS.2021.3113657.

[78] Zhang L, Zhang Q. Salient object detection with edge-guided learning and specific aggregation. IEEE Trans Circuits Syst Video Technol 2024;34(1):534–48. http://dx.doi.org/10.1109/TCSVT.2023.3287167.

[79] Zeng X, Xu M, Hu Y, Tang H, Hu Y, Nie L. Adaptive edge-aware semantic interaction network for salient object detection in optical remote sensing images. IEEE Trans Geosci Remote Sens 2023;61:1–16. http://dx.doi.org/10.1109/TGRS.2023.3300317.

[80] Zhou X, Shen K, Weng L, Cong R, Zheng B, Zhang J, Yan C. Edge-guided recurrent positioning network for salient object detection in optical remote sensing images. IEEE Trans Cybern 2023;53(1):539–52. http://dx.doi.org/10.1109/TCYB.2022.3163152.

[81] Chen J, Song Y, Li D, Lin X, Zhou S, Xu W. Specular removal of industrial metal objects without changing lighting configuration. IEEE Trans Ind Inf 2023.

[82] Xu H, Li Q, Chen J. Highlight removal from a single grayscale image using attentive GAN. Appl Artif Intell 2022;36(1):1988441.

[83] Zhang X, Zhou X, Lin M, Sun J. ShuffleNet: An extremely efficient convolutional neural network for mobile devices. In: IEEE/CVF conference on computer vision and pattern recognition. 2018, p. 6848–56. http://dx.doi.org/10.1109/CVPR.2018.00716.

[84] Iandola FN, Han S, Moskewicz MW, Ashraf K, Dally WJ, Keutzer K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size. 2016, arXiv preprint arXiv:1602.07360.

[85] Gao H, Wang Z, Ji S. ChannelNets: Compact and efficient convolutional neural networks via channel-wise convolutions. Adv Neural Inf Process Syst 2018;31.

[86] Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L-C. MobileNetV2: Inverted residuals and linear bottlenecks. In: IEEE/CVF conference on computer vision and pattern recognition. 2018, p. 4510–20. http://dx.doi.org/10.1109/CVPR.2018.00474.

[87] Zhang T, Qi G-J, Xiao B, Wang J. Interleaved group convolutions. In: IEEE international conference on computer vision. ICCV, 2017, p. 4383–92. http://dx.doi.org/10.1109/ICCV.2017.469.

[88] Xie G, Wang J, Zhang T, Lai J, Hong R, Qi G-J. Interleaved structured sparse convolutional neural networks. In: IEEE/CVF conference on computer vision and pattern recognition. 2018, p. 8847–56. http://dx.doi.org/10.1109/CVPR.2018.00922.

[89] Liu Z, Jiang T, Zhang T, Li Y. IR ship target saliency detection based on lightweight non-local depth features. In: The 3rd international conference on electronic information technology and computer engineering. EITCE, 2019, p. 1681–6. http://dx.doi.org/10.1109/EITCE47263.2019.9095005.

[90] Tu S, Rehman SU, Waqas M, Rehman OU, Shah Z, Yang Z, Koubaa A. ModPSO-CNN: an evolutionary convolution neural network with application to visual recognition. Soft Comput 2021;25:2165–76.

[91] Olaf R, Philipp F, Thomas B. U-Net: Convolutional networks for biomedical image segmentation. In: Medical image computing and computer-assisted intervention MICCAI international conference. 2015, p. 234–41.

[92] Chollet F. Xception: Deep learning with depthwise separable convolutions. In: IEEE conference on computer vision and pattern recognition. CVPR, 2017, p. 1800–7. http://dx.doi.org/10.1109/CVPR.2017.195.

[93] Zhou H, Xie X, Lai J-H, Chen Z, Yang L. Interactive two-stream decoder for accurate and fast saliency detection. In: IEEE conference on computer vision and pattern recognition. CVPR, 2020, p. 9138–47. http://dx.doi.org/10.1109/CVPR42600.2020.00916.

[94] Wei J, Wang S, Huang Q. F^3Net: Fusion, feedback and focus for salient object detection. In: Proceedings of the AAAI conference on artificial intelligence. Vol. 34, 2020, p. 12321–8.

[95] Chen T, Hu X, Xiao J, Zhang G. BPFINet: Boundary-aware progressive feature integration network for salient object detection. Neurocomputing 2021;451:152–66. http://dx.doi.org/10.1016/j.neucom.2021.04.078.

[96] Ke YY, Tsubono T. Recursive contour-saliency blending network for accurate salient object detection. In: 2022 IEEE/CVF winter conference on applications of computer vision. WACV, 2022, p. 1360–70. http://dx.doi.org/10.1109/WACV51458.2022.00143.

[97] Zhang J, Liang Q, Guo Q, Yang J, Zhang Q, Shi Y. R2Net: Residual refinement network for salient object detection. Image Vis Comput 2022;120:104423. http://dx.doi.org/10.1016/j.imavis.2022.104423.

[98] Deng B, French AP, Pound MP. Addressing multiple salient object detection via dual-space long-range dependencies. Comput Vis Image Underst 2023;235:103776. http://dx.doi.org/10.1016/j.cviu.2023.103776.

[99] 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. http://dx.doi.org/10.1109/TIM.2023.3290965.