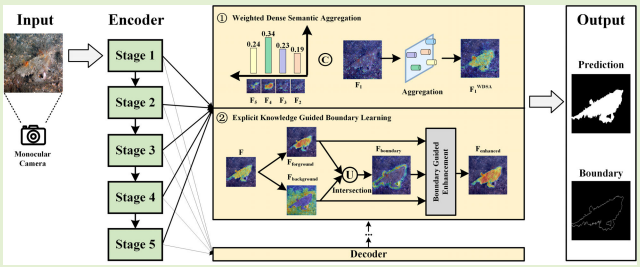
Weighted Dense Semantic Aggregation and Explicit Boundary Modeling for Camouflaged Object Detection

Weiyun Liang, *Student Member, IEEE*, Jiesheng Wu, *Student Member, IEEE*, Xinyue Mu, Fangwei Hao, Ji Du, Jing Xu, *Member, IEEE*, and Ping Li, *Member, IEEE*

***Abstract*—Camouflaged object detection (COD) in monocular images has garnered broad attention recently, aiming to segment objects that have high intrinsic similarity with their surroundings. Despite remarkable performance achieved by existing methods, two limitations persist: insufficient utilization of multilevel semantics at each decoding scale and a lack of "explicit" knowledge guidance in boundary learning, leading to performance drops in challenging scenarios. To address these issues, we propose a weighted dense semantic aggregation (WDSA) and explicit boundary modeling (EBM) network. Specifically, a WDSA module is proposed to sufficiently aggregate multilevel semantics at each decoding scale, and enable the exploration of the relationship between multilevel features and camouflaged objects. An EBM module is developed to capture edge semantics with explicit boundary knowledge guidance and enhance the feature representation with edge cues. A detail enhanced multiscale (DEMS) module is further designed to refine multiscale features. Extensive experiments demonstrate that our proposed method achieves competitive performance against state-of-the-art (SOTA) methods on four benchmark datasets without excessive model complexity. Codes and results will be released at https://github.com/crrcoo/SAE-Net.**



***Index Terms*— Boundary learning, camouflaged object detection (COD), weighted multilevel feature aggregation.**

I. Introduction

CAMOUFLAGE is a prevalent phenomenon widely observed in our daily lives, such as natural camouflage (e.g., animals mimic the environment to avoid predators) and artificial camouflage (e.g., military camouflage uniform) [1]. Camouflaged object detection (COD) is a dense prediction task, aiming to segment objects that have high intrinsic similarities with their surroundings [2]. As a fundamental segmentation task for objects in complex scenes, it can be applied to numerous downstream tasks such as autonomous driving [3], medical diagnosis [4], [5], surface defect detection [6], and underwater object detection [7]. Camouflage data typically consists of images or videos captured through monocular cameras [2], [8], [9]. Furthermore, some works combine monocular images with depth maps [10], [11] or use millimeter-wave images from active millimeter-wave scanners [12], [13], achieving promising results. Due to factors like equipment cost, obtaining these data in large quantities is often difficult, hindering their practical applications [10]. Hence, detecting camouflaged objects through easily accessible monocular images remains a critical issue, drawing broad attention within the computer vision community [14].

Compared to generic object detection (GOD) and salient object detection (SOD), COD mainly presents two additional challenges: the intrinsic similarity between foreground and background, and ambiguous boundary. In recent years, deep learning plays a crucial role in many fields of computer vision, such as object detection [15], semantic segmentation [16], and image recognition [17]. The multilevel feature representation capability makes deep models more effective than traditional methods in complex scenes. For the COD task, the multilevel contexts are prominent to understand the image content and distinguish camouflaged objects from the background environment. While existing COD methods [8], [18], [19], [20] have achieved remarkable performance, they typically follow

Manuscript received 1 April 2024; revised 9 May 2024; accepted 12 May 2024. Date of publication 22 May 2024; date of current version 1 July 2024. This work was supported in part by the Natural Science Foundation of Tianjin City under Grant 21JCYBJC00110. The associate editor coordinating the review of this article and approving it for publication was Prof. Xiaofeng Yuan. *(Corresponding author: Jing Xu.)*

Weiyun Liang, Jiesheng Wu, Xinyue Mu, Fangwei Hao, Ji Du, and Jing Xu are with the College of Artificial Intelligence, Nankai University, Tianjin 300350, China (e-mail: weiyunliang@mail.nankai.edu.cn; jasonwu@mail.nankai.edu.cn; muxinyue@mail.nankai.edu.cn; haofangwei@mail.nankai.edu.cn; 1120230244@mail.nankai.edu.cn; xujing@n-ankai.edu.cn).

Ping Li is with the Department of Computing and the School of Design, The Hong Kong Polytechnic University, Hong Kong (e-mail: p.li@polyu.edu.hk).

Digital Object Identifier 10.1109/JSEN.2024.3401722

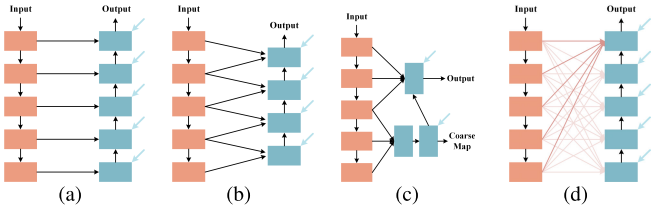


Fig. 1. Different types of decoding structures for COD. (a) FPN-based structure [1], [23], [24], [28]. (b) Adjacent level fusion structure [25], [26]. (c) Low- and high-level feature separate decoding structure [2], [27]. (d) Our proposed weighted dense aggregation structure. Orange and blue blocks are encoder and decoder modules. Blue arrows denote supervisions. Red arrows in (d) represent our proposed ILS and ILF steps applied to multilevel features.

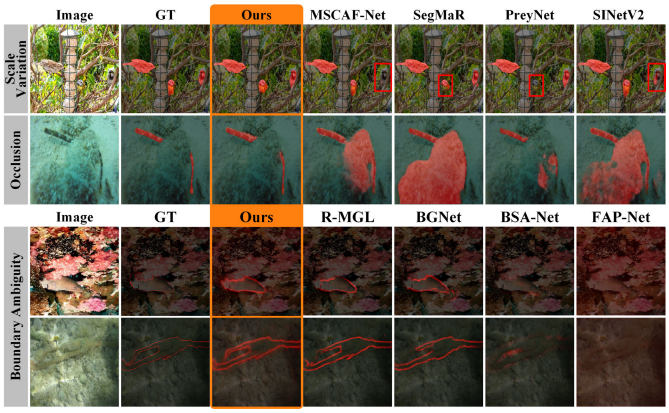


Fig. 2. Visual comparison of COD in challenging scenarios, i.e., scale variation (row 1), occlusion (row 2), and boundary ambiguity (rows 3 and 4). Our method achieves superior performance with more complete results and fewer false alarms compared to MSCAF-Net [32], SegMaR [33], PreyNet [26], and SINetV2 [1], and generates more complete edges compared to R-MGL [34], BGNet [35], BSA-Net [36], and FAP-Net [25].

an encoder–decoder architecture. The sufficient utilization of multilevel features extracted from the encoder remains a key factor in determining the effectiveness of these methods. Feature pyramid network (FPN) [15] is a decoding structure adept at effectively utilizing multilevel features, which has been widely applied in tasks such as GOD [21] and SOD [22]. In the context of COD, several works [18], [19], [23], [24] adopt an FPN-based structure, as illustrated in Fig. 1(a). Taking it further, some FPN-based variants are proposed to obtain richer multilevel semantics during decoding phase, e.g., fusing features from adjacent levels [25], [26] and separately decoding high- and low-level features [2], [27] as shown in Fig. 1(b) and (c), respectively. In a word, the principle advantage of these FPN-based structure lies in its ability to combine semantically strong, low-resolution features with semantically weak, high-resolution features in a top-down manner, ultimately obtaining semantically strong and high-resolution features at the network output [15].

Intuitively, due to the challenges in COD, relying on capturing rich multilevel semantics at the output decoding stage may not be sufficient for effectively distinguishing camouflaged objects from background. Despite the notable performance achieved by the aforementioned FPN-based methods, there are still three issues worth investigating. First, since high- and low-level features do not directly interact, the model struggles to fully utilize multilevel semantics in the early and intermediate stages. This might lead to incomplete detection in complex scenes with multiple targets, e.g., missing some target birds as shown in the first row of Fig. 2. Second, the top-down decoding process weakens high-level features, leading to a lack of global semantics in the network output. This may cause over-segmentation, e.g., misclassifying background as foreground as shown in the second row of Fig. 2. Third, these multilevel decoding structures are empirically designed to integrate limited multilevel semantics at each decoding stage, and the relationship between multilevel features and camouflaged objects has not been thoroughly investigated.

To address the above three issues, we observe that the dense connection strategy [29] empowers each decoding stage with the potential to fully acquire multilevel semantics. Also, this strategy can be considered a general form of various FPN variants, with the potential to explore the relationship between multilevel features and camouflaged objects. However, only a limited number of works have explored the use of dense connection or its variants for the COD task. The effective utilization of dense connection-related decoding structures for COD still needs further investigation. Directly applying dense connection to COD poses two problems. First, despite the complementary information in multilevel features, there are still semantic gaps between these features [30], [31]. Naive combinations of these features may dilute the impact of strong cues with weak signals or even amplify the adverse effects of confusing responses. Second, fusing features from all levels at each stage may cause model redundancy.

In response to these problems, we propose a weighted dense semantic aggregation (WDSA) module comprising of an intra-level selection (ILS) step and an inter-level fusion (ILF) step, as shown in Fig. 1(d). The ILS step obtains complementary multilevel semantic features tailored to each decoding scale, filtering out confusing and redundant features. The ILF step considers the distinct contributions of various levels to a specific decoding scale, and adaptively interacts multilevel features through a weighting strategy. With this design, the WDSA module effectively enhances multilevel features at each decoding scale with sufficient multilevel semantics. As illustrated in Fig. 3(b), the output of the WDSA ($\mathbf{F}_1^{\mathrm{WDSA}}$) integrates both the global location information from high-level features and texture details from low-level features. Besides, analyzing the weights of the WDSA facilitates a quantitative exploration of the correlation between multilevel features and camouflaged objects, an aspect that has not been fully investigated.

In addition, camouflaged objects hidden in complex scenes often exhibit low contrast between their interiors and environment, leading to ambiguous boundaries. Existing methods often draw on boundary learning approaches similar to SOD models [37], [38]. They typically utilize a module [31], [35] or a sub-network [25], [36] to extract boundaries from backbone features. Furthermore, [39] simultaneously models foreground, background, and edge semantics to better learn each semantic. However, the boundary learning in the aforementioned methods primarily relies on edge supervision of image features, which typically contain weak edge priors. This

"implicit" boundary learning approach might not adequately model edge semantics for the COD task with ambiguous edges. As shown in the third and fourth rows of Fig. 2, recent boundary-related methods might struggle to generate intact edge contours [34], [35] (e.g., col. 4 and 5) or detect clear edges [25], [36] (e.g., col. 6 and 7). Therefore, we consider to incorporate "explicit" boundary knowledge guidance to facilitate the boundary learning. Motivated by the observation that boundary serves as the separator between foreground and background [36], we propose an explicit boundary modeling (EBM) module to "explicitly" obtain boundary features from regions jointly attended by foreground and background features. As illustrated in Fig. 3(c), compared to image features ($\mathbf{F}$), the boundary features ($\mathbf{F}_{\text{boundary}}$) contain strong edge priors, making it easier to learn intact and continuous edges. Then the EBM module adopts boundary features as attentional guidance to guide features focusing on boundary details.

Based on the above insights and investigations, we propose a weighted dense semantic aggregation and explicit boundary modeling network for COD, namely SAE-Net, that incorporates both WDSA and EBM modules. Besides, to refine the multilevel features and increase the receptive field in the decoding phase as commonly done in [2], [32], and [36], we further design a detail-enhanced multiscale (DEMS) module to extract refined multiscale features with subtle cues. Our main contributions are summarized as follows.

1) We propose a COD framework named SAE-Net via employing weighted dense multilevel semantic aggregation at each decoding scale and capturing edge semantics with an explicit boundary knowledge guidance. The weighted dense aggregation assists in exploring the correlation between multilevel features and camouflaged objects, which has not been thoroughly investigated.
2) A WDSA module is proposed to enhance multilevel semantics at each decoding scale, enabling the adaptive selection and fusion of complementary multilevel features. A DEMS module is further designed to extract refined multiscale features with subtle cues.
3) An EBM module is developed to capture edge semantics through explicit boundary knowledge guidance and enhance the feature representation with edge cues.
4) Extensive experiments demonstrate the proposed SAE-Net achieves competitive against state-of-the-art (SOTA) methods on four benchmark datasets without excessive model complexity. Visualizations further illustrate the superior performance of our model on various challenging cases.

II. Related Work

A. Camouflaged Object Detection

Research on COD has significantly advanced our understanding of visual perception and has attracted intensive attention from the computer vision community [14]. Since traditional methods often struggle to deal with complex scenarios, recent works mainly adopt convolutional neural network (CNN)-based or transformer-based methods to address this problem. For example, Fan et al. [2] propose SINet and SINetV2 [1] for COD that simulate the search and identify stages of a predator's hunting process. Lv et al. [8] introduce a new camouflaged object ranking task and the CAM-FR dataset with both localization and ranking annotations. Pang et al. [40] learn the discriminative mixed-scale semantics by a zoom strategy. Jia et al. [33] integrate segment, magnify, and reiterate in a multistage detection fashion for accurate COD. Zhong et al. [41] and He et al. [19] adopt frequency-domain features to assist COD. Hu et al. [42] exploit the high-resolution information to refine the low-resolution representations via iterative feedback. Song et al. [43] detect camouflaged objects by generating focus areas. Xing et al. [44] propose a search-amplify-recognize architecture that draws inspiration from humans tend to go closer to see the ambiguous objects. Although existing methods achieve good performance, there are still two problems. First, the FPN-based frameworks often insufficiently fuse multilevel features at each decoding scale. Second, the boundary-related methods lack "explicit" guidance in boundary learning. Our proposed method aims to deal with the two issues.

B. Context-Aware Feature Learning

Contextual information is able to enhance the feature representation and has shown its effectiveness in object detection and segmentation tasks [32], [45]. Many works are dedicated to exploit rich contextual information to facilitate the feature learning, such as [46], [47], [48], [49], and [50]. For the COD task, on the one hand, some works integrate features from multiple stages to utilize multilevel contexts. For instance, Zhang et al. [26] propose a bidirectional bridging interaction module to select and aggregate features of adjacent stages in an attentive manner. Chen et al. [27] separately decode high- and low-level features to generate an initial prediction and the final prediction. Zhou et al. [25] develop a cross-level fusion and propagation module that integrates features from adjacent layers to exploit cross-level correlations. On the other hand, some works exploit multiscale contexts in each stage. For example, SINetV2 [1] adopts a receptive field block (RFB), which consists of parallel branches with different dilation rates to extract discriminative multiscale features. Zhu et al. [36] further integrate the multiscale features between parallel branches to enrich the multiscale feature representation. Liu et al. [32] design an enhanced receptive field (ERF) module to learn rich contexts. However, these methods are typically based on an FPN structure and insufficiently utilize multilevel semantics, resulting in performance degradation in complex scenes. Different from the above works, we propose a WDSA module to effectively enhance multilevel semantics at each decoding scale and a DEMS module to obtain refined multiscale features with detail cues. Besides, visualizing the weights of WDSA modules assists in exploring the correlation between multilevel features and camouflaged objects, which has not been thoroughly investigated.

C. Boundary Learning

Despite COD, the boundary learning has been widely used in other computer vision tasks, e.g., SOD [37], [38], polyp

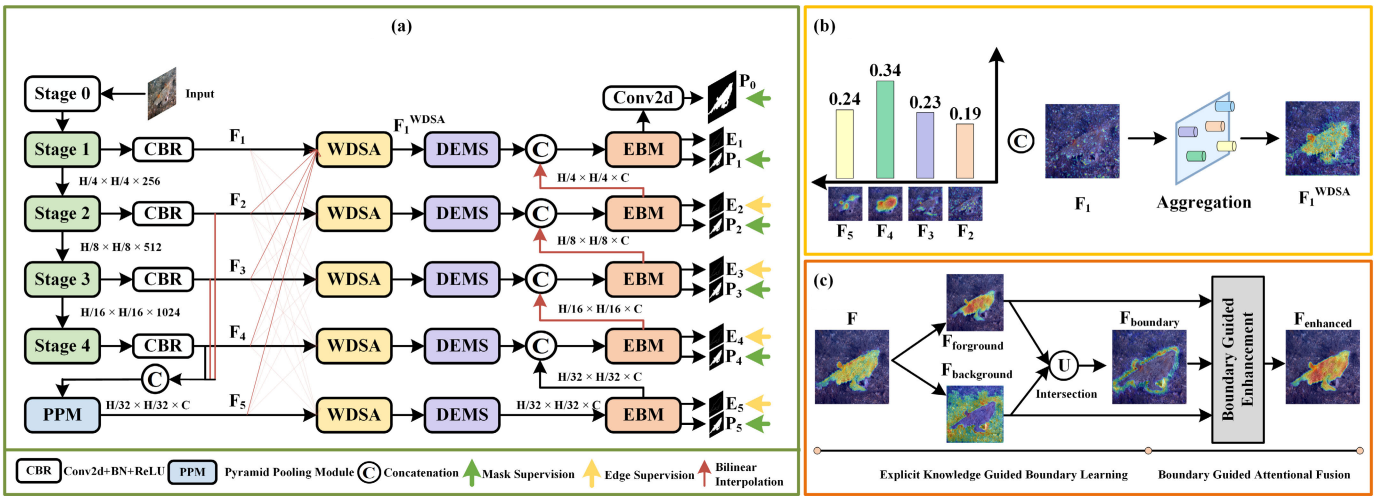


Fig. 3. (a) Overall architecture of our proposed SAE-Net. It consists of three main components, i.e., WDSA, DEMS, and EBM. (b) Simplified pipeline of WDSA. WDSA adaptively aggregates multilevel semantics to sufficiently integrate the global localization information of high-level features and the texture information of low-level features for each distinct decoding scale. (c) Simplified pipeline of EBM. The boundary features ($\mathbf{F}_{\text{boundary}}$) learned through explicit boundary knowledge guidance contain strong edge priors compared to image features ($\mathbf{F}$), making it easier to learn intact and continuous edges.

segmentation [51], and semantic segmentation [52]. For COD task, Zhai et al. [34] adopt edge-constricted graph reasoning to guide the feature representation learning by edge information. Ji et al. [31] extract edge from low-level features and utilize edge priors to guide the coarse map refinement. Wu et al. [39] learn foreground, background, and edge semantics simultaneously to facilitate the boundary learning. Zhu et al. [36] design a simple network to detect boundary that is further embedded into the feature maps. Sun et al. [35] utilize both low-level and high-level features to generate edge feature that is used to enhance the feature representation. Zhou et al. [25] extract edges from low-level features, which are adopted to enhance the detection in the decoding procedure. Unlike prior works that "implicitly" extract boundary from image features using edge supervision, we incorporate an explicit boundary knowledge guidance to assist the edge semantic learning.

III. Proposed Method

A. Overview

The overall architecture of our proposed method is shown in Fig. 3. Specifically, the Res2Net-50 [53] backbone extracts hierarchical features from an input RGB image $\mathbf{I} \in \mathbb{R}^{H\times W\times 3}$, where H and W represent the height and width, respectively. Then, CBR blocks reduce the feature channel to C, where the CBR block consists of a convolutional layer, a batch normalization layer [54], and a ReLU activation. The output features are denoted as $\{\mathbf{F}_i\}_{i=1}^{4}$, where $\mathbf{F}_i \in \mathbb{R}^{H_i\times W_i\times C}$ and i indicates the ith stage. Moreover, we incorporate a pyramid pooling module (PPM) [55] on top of the backbone to capture global semantics, as done in [56], [57], and [58]. WDSA and DEMS modules are then sequentially employed to explore multilevel features and refined multiscale features at each decoding stage. Finally, EBM modules gradually generate boundary maps and the final prediction in a top-down manner. Bilinear interpolation is used throughout the network to keep feature sizes consistent if necessary.

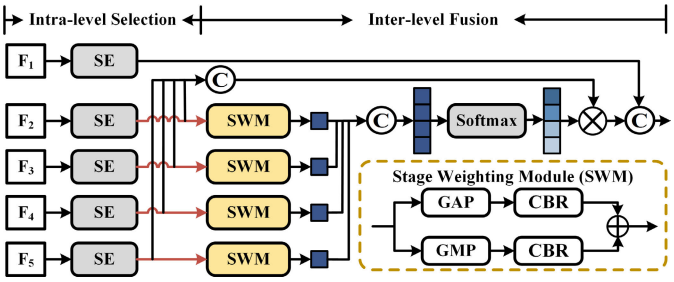


Fig. 4. Architecture of the WDSA module for the first stage. SE represents the squeeze-and-excitation block [59].

B. WDSA Module

It is known that multilevel features typically encode distinct semantics and focus on different object scales [30]. However, the decoding stages of the existing FPN-based methods [18], [19] often integrate limited multilevel features, hindering the sufficient utilization of multilevel semantics to distinguish camouflaged objects from background for the COD task. Therefore, the WDSA module is proposed to adaptively leverage multilevel semantics at each decoding scale. Considering the feature inconsistency and redundancy in multilevel features [15], [31], the WDSA module consists of an ILS step and an ILF step. ILS obtains complementary multilevel semantics specific to that decoding scale. Then, ILF weights and fuses multilevel features by considering the diverse contributions of different levels to the designated decoding scale. The experimental comparison between WDSA and other multilevel aggregation strategies is discussed in Section IV-E2. Feature visualizations of the WDSA module are shown in Section IV-E5.

In this section, we will introduce the WDSA module in the first stage as an illustrative example to ensure clarity. The architecture of our WDSA module in the first stage is shown in Fig. 4. The inputs of the WDSA module in the first stage are denoted as $\{\mathbf{F}_j\}_{j=1}^{5}$.

1) Intra-Level Selection: Attention mechanism is commonly used in computer vision to select useful features. Motivated by

this, we employ it to filter out semantically redundant information between multilevel features to mitigate their semantic gap. We use the classic squeeze-and-excitation (SE) block [59] to filter the inconsistent semantics of the multilevel features. Motivated by the vanilla FPN [15], ILS considers $\mathbf{F}_1$ as the primary feature of the first stage, where the resolution of $\mathbf{F}_1$ matches the feature scale of the first stage. $\{\mathbf{F}_j\}_{j=2}^{5}$ are regarded as auxiliary features to complement $\mathbf{F}_1$. The ILS reduces the channel of $\mathbf{F}_1$ to C_m to reduce the feature redundancy and retain the unique characteristics of the current feature scale. The channels of $\{\mathbf{F}_j\}_{j=2}^{5}$ are reduced to C_{aux} to filter out redundant information in features with inconsistent semantics. The ILS is formulated by

$$\mathbf{F}_j^{\text{ILS}} = \text{Up}(\text{SE}(\mathbf{F}_j)), \quad j \in \{1, 2, 3, 4, 5\} \tag{1}$$

where $\text{Up}(\cdot)$ denotes the bilinear interpolation operation. $\mathbf{F}_j^{\text{ILS}}$ denotes the output of the ILS for $\mathbf{F}_j$, and the resolution of $\mathbf{F}_j^{\text{ILS}}$ is $H/4 \times W/4$. We compare SE with other commonly used attention modules in Section IV-E2. The chosen of C_m and C_{aux} will be discussed in Section IV-E3.

2) Inter-Level Fusion: Directly fusing multilevel features assumes that features of different levels have an equal impact at different decoding stages. This prevents the network from adaptively selecting prominent features to fit specific feature scales. Therefore, the ILF is proposed to adaptively assign weights for multilevel features to highlight the features that are more relevant to a specific scale.

In the ILF, a stage weighting module (SWM) is applied to each auxiliary features to generate scale values $\{v_j | j \in \{2, 3, 4, 5\}\}$ that represent the importance of the auxiliary features. The SWM is formulated as

$$\begin{aligned} v_j &= f_{1\times1}\left(\text{GAP}\left(\mathbf{F}_j^{\text{ILS}}\right)\right) \oplus f_{1\times1}\left(\text{GMP}\left(\mathbf{F}_j^{\text{ILS}}\right)\right) \\ \{w_j\}_{j=2}^{5} &= \text{softmax}\left(\{v_j\}_{j=2}^{5}\right) \end{aligned} \tag{2}$$

where $\{w_j\}_{j=2}^{5}$ are the weights for $\{\mathbf{F}_j^{\text{ILS}}\}_{j=2}^{5}$ and $\oplus$ denotes the element-wise addition. $f_{1\times1}$ denotes the CBR block with 1×1 convolution. $\text{softmax}(\cdot)$ indicates the *softmax* operation for the input values. The SWM utilizes both max and average pooling operations because they concentrate on different types of information and are complementary to each other [60], [61]. The final output of the WDSA module can be computed by

$$\begin{aligned} \dot{\mathbf{F}}_j^{\text{ILS}} &= w_j \otimes \mathbf{F}_j^{\text{ILS}}, \quad j \in \{2, 3, 4, 5\} \\ \mathbf{F}_1^{\text{WDSA}} &= \text{Concat}\left[\mathbf{F}_1^{\text{ILS}}; \dot{\mathbf{F}}_2^{\text{ILS}}; \dot{\mathbf{F}}_3^{\text{ILS}}; \dot{\mathbf{F}}_4^{\text{ILS}}; \dot{\mathbf{F}}_5^{\text{ILS}}\right] \end{aligned} \tag{3}$$

where $\text{Concat}[\cdot;\cdot]$ represents the concatenation operation and $\otimes$ indicates the element-wise multiplication. $\mathbf{F}_1^{\text{WDSA}}$ is the output of the WDSA module in the first stage. $\{\mathbf{F}_i^{\text{WDSA}}\}_{i=1}^{5}$ will be fed into DEMS modules to further explore refined multiscale contexts. In Section IV-E4, we visualize the weights of multilevel features and explore their relationship with camouflaged objects.

C. DEMS Module

Existing COD methods [1], [2], [33] commonly utilize dilated convolutions with varying dilation rates to expand the receptive field. Since the high similarity between foreground and background, discriminative camouflaged traces are subtle and difficult to capture [41], [62]. However, dilated convolutions (DConvs) inherently contain dilation gaps, ignoring the accompanying loss of intricate details. This may cause existing multiscale modules to struggle to capture subtle clues for the COD task. Motivated by this, we design a DEMS module that incorporates the detail-enhanced convolution (DeConv) to refine the fused multilevel features. The experimental comparison between DEMS and other multiscale modules, i.e., atrous spatial pyramid pooling (ASPP) [63], RFB [1], and ERF [32], is discussed in Sections IV-E6 and IV-E7.

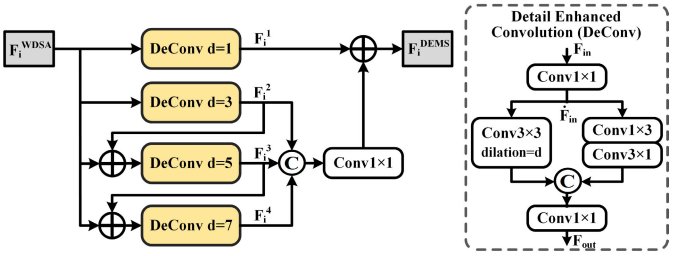


Fig. 5. Architecture of the DEMS module.

The architecture of the DEMS module is illustrated in Fig. 5. The DeConv block adopts asymmetric convolution (AConv) to assist the DConv with detailed features. Denote the input of the DeConv block as $\mathbf{F}_{\text{in}}$, the DeConv block can be computed by

$$\begin{aligned} \dot{\mathbf{F}}_{\text{in}} &= g_{1\times1}(\mathbf{F}_{\text{in}}) \\ \mathbf{F}_{\text{out}} &= g_{1\times1}(\text{Concat}[h_d(\dot{\mathbf{F}}_{\text{in}}); g_{3\times1}(g_{1\times3}(\dot{\mathbf{F}}_{\text{in}})]) \end{aligned} \tag{4}$$

where $g_{k_1\times k_2}$ denotes the $k_1 \times k_2$ convolutional layer. h_d denotes the 3×3 convolutional layer with dilation rate d. $\dot{\mathbf{F}}_{\text{in}}$ and $\mathbf{F}_{\text{out}}$ are an intermediate feature and the output of the DeConv block, respectively.

The input of the DEMS module is $\mathbf{F}_i^{\text{WDSA}}$. We denote $\{\mathbf{F}_i^z\}_{z=1}^{4}$ as the outputs of the four branches, and they can be computed by

$$\begin{aligned} \mathbf{F}_i^1 &= \text{DeConv}_1\left(\mathbf{F}_i^{\text{WDSA}}\right) \\ \mathbf{F}_i^2 &= \text{DeConv}_3\left(\mathbf{F}_i^{\text{WDSA}}\right) \\ \mathbf{F}_i^3 &= \text{DeConv}_5\left(\mathbf{F}_i^{\text{WDSA}} \oplus \mathbf{F}_i^2\right) \\ \mathbf{F}_i^4 &= \text{DeConv}_7\left(\mathbf{F}_i^{\text{WDSA}} \oplus \mathbf{F}_i^3\right) \\ \mathbf{F}_i^{\text{DEMS}} &= g_{1\times1}\left(\text{Concat}\left[\mathbf{F}_i^2; \mathbf{F}_i^3; \mathbf{F}_i^4\right]\right) \oplus \mathbf{F}_i^1 \end{aligned} \tag{5}$$

where DeConv_d denotes the DeConv block with dilation rate d. $\mathbf{F}_i^{\text{DEMS}}$ is the output of DEMS, which encodes refined multiscale information and will be sent into the EBMs.

D. EBM Module

Existing boundary-related methods [25], [31], [35], [36], [39] often rely solely on "implicit" edge supervision of image features, which typically contains weaker edge priors. These "implicit" approaches might not adequately model edge semantics for the COD task with ambiguous edges. Therefore, we consider to adopt an "explicit" boundary guidance to facilitate the boundary semantic learning. Motivated by the observation that the boundary serves as the separator between the foreground and background [36], we propose an EBM

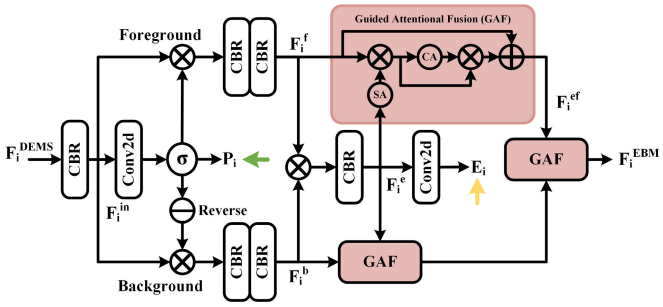


Fig. 6. Architecture of the EBM module. $\mathbf{P}_i$ and $\mathbf{E}_i$ are supervised by mask and boundary ground truths, respectively.

module to learn boundary from features jointly attended by foreground and background features, and enhance the feature representation with edge semantics. The experimental comparison between EBM and existing boundary learning methods, i.e., SEA [36] and MECSA [39], is discussed in Sections IV-E8 and IV-E9. The architecture of the EBM is shown in Fig. 6, comprising explicitly knowledge-guided boundary learning (EKGBL) and boundary-guided attentional fusion (GAF) procedures.

1) Explicitly Knowledge-Guided Boundary Learning: In addition to edge supervision, the key procedure in edge generation is the EKGBL, which consists of two main steps. First, it extracts foreground feature $\mathbf{F}_i^f$ and background feature $\mathbf{F}_i^b$ separately through mask supervision. Second, it models boundary feature $\mathbf{F}_i^e$ as commonly interested regions of foreground and background features by multiplying $\mathbf{F}_i^f$ and $\mathbf{F}_i^b$. Denote $\mathbf{F}_i^{\text{in}}$ as an intermediate feature, $\mathbf{F}_i^e$ can be computed by

$$\begin{aligned}
\mathbf{F}_i^{\text{in}} &= f_{3\times3}\left(\mathbf{F}_i^{\text{DEMS}}\right)\\
\mathbf{P}_i &= \sigma\left(\text{Conv}_{1\times1}\left(\mathbf{F}_i^{\text{in}}\right)\right)\\
\mathbf{F}_i^f &= f_{3\times3}\left(f_{3\times3}\left(\mathbf{P}_i \otimes \mathbf{F}_i^{\text{in}}\right)\right)\\
\mathbf{F}_i^b &= f_{3\times3}\left(f_{3\times3}\left((1-\mathbf{P}_i) \otimes \mathbf{F}_i^{\text{in}}\right)\right)\\
\mathbf{F}_i^e &= f_{3\times3}\left(\mathbf{F}_i^f \otimes \mathbf{F}_i^b\right)\\
\mathbf{E}_i &= \text{Conv}_{1\times1}\left(\mathbf{F}_i^e\right)
\end{aligned} \tag{6}$$

where σ denotes the *sigmoid* function and $\text{Conv}_{k\times k}$ indicates the $k \times k$ convolutional layer. $\mathbf{P}_i$ and $\mathbf{E}_i \in \mathbb{R}^{H_i\times W_i\times 1}$ are the coarse mask prediction and boundary prediction, respectively. They are supervised by mask and boundary ground truths.

2) Boundary GAF: Based on the previous study [31], boundary features can mitigate the edge vanishing problem and preserve finer structure details. Inspired by this and [70], [71], we design a GAF module to enhance the feature representation. For instance, given the inputs $\mathbf{F}_i^f$ and $\mathbf{F}_i^e$, the GAF module first performs spatial attention (SA) on $\mathbf{F}_i^e$. Next, the SA map is multiplied with $\mathbf{F}_i^f$ and we get an intermediate feature $\mathbf{F}_i^m \in \mathbb{R}^{H_i\times W_i\times C}$. Then, the channel attention (CA) is applied to $\mathbf{F}_i^m$ and the CA map is multiplied with $\mathbf{F}_i^m$. Finally, a residual flow is further employed to enhance the feature representation. The GAF module is formulated as

$$\begin{aligned}
\mathbf{F}_i^m &= \mathbf{F}_i^f \otimes \sigma\left(\text{Conv}_{1\times1}\left(\mathbf{F}_i^e\right)\right)\\
\mathbf{F}_i^{\text{ef}} &= \mathbf{F}_i^f \oplus \left(\mathbf{F}_i^m \otimes s\left(\text{Conv}_{1\times1}\left(\text{GAP}\left(\mathbf{F}_i^m\right)\right)\right)\right)
\end{aligned} \tag{7}$$

where s is the *softmax* operation along the channel dimension. $\mathbf{F}_i^{\text{ef}} \in \mathbb{R}^{H_i\times W_i\times C}$ is the enhanced foreground feature. Here, the GAF adopts boundary feature as additional guidance to guide foreground feature focusing on the boundary regions, thereby enhancing the foreground feature. EBM adopts three GAF modules to fuse foreground, background, and boundary features. This procedure is defined as

$$\mathbf{F}_i^{\text{EBM}} = \text{GAF}\left(\mathbf{F}_i^{\text{ef}}, \text{GAF}\left(\mathbf{F}_i^b, \mathbf{F}_i^e\right)\right) \tag{8}$$

where $\mathbf{F}_i^{\text{EBM}} \in \mathbb{R}^{H_i\times W_i\times C}$ is the output of the EBM module. $\mathbf{F}_i^{\text{EBM}}$ will be fed into the next stage to gradually generate the final prediction in a top-down manner.

E. Loss Function

Following [1], [2], [25], [27], [32], [33], [35], [68], we adopt the structure loss (L_{str}) [72] for mask supervision. Besides, since the proportion of the edge pixels is typically low in an image, the focal loss [73] (L_{focal}) is adopted for boundary supervision. The final prediction of our model is $\mathbf{P}_0$. We add side supervisions to the coarse prediction maps $\{\mathbf{P}_i\}_{i=1}^5$ and the boundary prediction maps $\{\mathbf{E}_i\}_{i=1}^5$. Following [1], [2], these predicted maps are upsampled to the same size as the ground truths using bilinear interpolation and then used to compute the loss alongside the ground truths. Denote mask and boundary ground truths as $\mathbf{G}_m$ and $\mathbf{G}_e$, respectively, the loss function is defined as

$$L = \sum_{i=0}^{5} L_{\text{str}}(\text{Up}(\mathbf{P}_i), \mathbf{G}_m) + \sum_{i=1}^{5} L_{\text{focal}}(\text{Up}(\mathbf{E}_i), \mathbf{G}_e) \tag{9}$$

where $\text{Up}(\cdot)$ indicates the bilinear interpolation operation.

IV. Experiments

A. Datasets and Evaluation Metrics

To demonstrate the effectiveness of our model, we conduct experiments on four benchmark datasets. **CHAMELEON** [74] contains 76 camouflaged images and used for testing. **CAMO** [75] is composed of 1250 camouflaged images, and is divided into a training set of 1000 images and a test set of 250 images. **COD10K** [2] contains 5066 camouflaged images. It has been pre-divided into a training set of 3040 images and a test set of 2026 images. **NC4K** [8] contains 4121 camouflaged images. It is commonly used as a test set to evaluate the generalization capability of the model. Following [1], [2], the combination of the training sets of CAMO and COD10K is adopted for training. Other data is utilized for testing. The boundary ground truths are provided by [35].

Following [1], [2], we adopt four commonly used evaluation metrics: structure-measure (S_α) [76], mean E-measure (E_ϕ) [77], weighted F-measure (F_β^w) [78], and mean absolute error (MAE) [79]. All the metrics are calculated by the evaluation toolbox provided by [2].

B. Training and Implementation Details

Following [1], [25], [35], [36], [68], the Res2Net-50 [53] pre-trained on ImageNet [17] is adopted as the backbone network. Following [25], [26], [27], [32], [36], we set C

TABLE I
COMPARISON WITH CNN-BASED METHODS. † DENOTES THE RESULTS ARE DIRECTLY CITED FROM THE ORIGINAL PAPERS. – DENOTES THE RESULTS ARE UNAVAILABLE

Method	Publication	Backbone	CHAMELEON (76)				CAMO (250)				COD10K (2,026)				NC4K (4,121)			
			$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$M \downarrow$
SINet [2]	CVPR'20	ResNet-50	0.869	0.891	0.740	0.044	0.751	0.771	0.606	0.100	0.771	0.806	0.551	0.051	0.808	0.871	0.723	0.058
PFNet [23]	CVPR'21	ResNet-50	0.882	0.931	0.810	0.033	0.782	0.841	0.695	0.085	0.800	0.877	0.660	0.040	0.829	0.887	0.745	0.053
R-MGL [34]	CVPR'21	ResNet-50	0.893	0.917	0.812	0.031	0.776	0.812	0.673	0.088	0.814	0.851	0.666	0.035	0.833	0.867	0.739	0.053
LSR [8]	CVPR'21	ResNet-50	0.890	0.935	0.822	0.030	0.787	0.838	0.696	0.080	0.804	0.880	0.673	0.037	0.840	0.895	0.766	0.048
UGTR [64]	ICCV'21	ResNet-50	0.888	0.911	0.796	0.031	0.785	0.823	0.686	0.086	0.818	0.853	0.667	0.035	0.839	0.874	0.747	0.052
JCOD [65]	CVPR'21	ResNet-50	0.891	0.945	0.833	0.030	0.800	0.859	0.728	0.073	0.809	0.884	0.684	0.035	0.842	0.898	0.771	0.047
C^2F-Net [30]	IJCAI'21	Res2Net-50	0.888	0.935	0.828	0.032	0.796	0.854	0.719	0.080	0.813	0.890	0.686	0.036	0.838	0.897	0.762	0.049
ERRNet [31]	PR'22	ResNet-50	0.868	0.922	0.787	0.039	0.779	0.842	0.679	0.085	0.786	0.867	0.630	0.043	0.827	0.887	0.737	0.054
CubeNet [66]	PR'22	ResNet-50	0.873	0.928	0.786	0.037	0.788	0.838	0.682	0.085	0.795	0.865	0.643	0.041	-	-	-	-
FBNet† [67]	TOMM'22	ResNet-50	0.888	0.939	0.828	0.032	0.783	0.839	0.702	0.081	0.809	0.889	0.684	0.035	-	-	-	-
DTC-Net† [24]	TMM'22	ResNet-50	0.876	0.897	0.773	0.039	0.778	0.804	0.667	0.084	0.790	0.821	0.616	0.041	-	-	-	-
SegMaR [33]	CVPR'22	ResNet-50	**0.906**	0.951	**0.860**	**0.025**	0.815	0.874	0.753	0.071	0.833	0.899	0.724	0.034	0.841	0.896	0.781	0.046
SINetV2 [1]	TPAMI'22	Res2Net-50	0.888	0.942	0.816	0.030	0.820	0.882	0.743	0.070	0.815	0.887	0.680	0.037	0.847	0.903	0.770	0.048
C^2F-NetV2 [27]	TCSVT'22	Res2Net-50	0.893	0.946	0.845	0.028	0.799	0.859	0.730	0.077	0.811	0.887	0.691	0.036	0.840	0.896	0.770	0.048
PreyNet [26]	MM'22	Res2Net-50	0.895	**0.952**	0.844	0.028	0.790	0.842	0.708	0.077	0.813	0.881	0.697	0.034	0.834	0.887	0.763	0.050
BSA-Net [36]	AAAI'22	Res2Net-50	0.895	0.946	0.841	0.027	0.794	0.851	0.717	0.079	0.818	0.891	0.699	0.034	0.841	0.897	0.771	0.048
FindNet† [68]	TIP'22	Res2Net-50	0.895	0.944	0.839	0.027	0.080	0.862	0.725	0.077	0.811	0.883	0.688	0.036	0.841	0.895	0.769	0.048
FAP-Net [25]	TIP'22	Res2Net-50	0.893	0.940	0.825	0.028	0.815	0.865	0.734	0.076	0.822	0.888	0.694	0.036	0.851	0.899	0.775	0.047
BGNet [35]	IJCAI'22	Res2Net-50	0.901	0.943	0.850	0.027	0.812	0.870	0.749	0.073	0.831	0.901	0.722	0.033	0.851	0.907	0.788	0.044
R-MGL_V2† [69]	TIP'23	ResNet-50	0.892	0.935	0.825	0.029	0.774	0.848	0.684	0.085	0.813	0.882	0.682	0.034	0.831	0.892	0.751	0.051
PUENet [20]	TIP'23	Res2Net-50	0.879	0.928	0.818	0.036	0.794	0.861	0.722	0.077	0.809	0.884	0.687	0.037	0.835	0.892	0.762	0.049
MSCAF-Net [32]	TCSVT'23	Res2Net-50	0.880	0.926	0.800	0.033	0.800	0.855	0.717	0.077	0.808	0.879	0.667	0.038	0.855	0.908	0.792	0.043
MRR-Net [28]	TNNLS'23	Res2Net-50	0.891	0.936	0.819	0.029	0.826	0.880	0.759	0.070	0.835	0.901	0.720	0.032	0.857	0.906	0.786	0.044
SAE-Net (Ours)	-	Res2Net-50	0.896 (±0.0026)	0.949 (±0.0034)	0.833 (±0.0079)	0.027 (±0.0009)	**0.837** (±0.0009)	**0.891** (±0.0003)	**0.770** (±0.0033)	**0.064** (±0.0008)	**0.838** (±0.0029)	**0.904** (±0.0027)	**0.727** (±0.0074)	**0.031** (±0.0007)	**0.862** (±0.0012)	**0.912** (±0.0005)	**0.796** (±0.0044)	**0.042** (±0.0005)

as 64. In the training process, input images are resized to 384 × 384 following [19], [28], [36], [39], [40], [43], [80], [81], and augmented by random horizontal flipping and color enhancement. We utilize the Adam optimizer [82] with a weight decay of 5e-4. The initial learning rate is set to 1e-4, which will decay 0.5 times every 10 epochs. The total training epoch is 100. Our model is implemented with PyTorch framework and trained on a single NVIDIA RTX 3090 GPU. The batch size is set as 20. In the testing phase, the input image is resized to 384 × 384, and the output prediction map is then resized back to the original image size for calculating the evaluation metrics with the ground truths. We do not adopt any post-processing to enhance our final prediction.

C. Comparison to SOTAs

We compare our SAE-Net against 23 SOTA methods, i.e., SINet [2], PFNet [23], UGTR [64], R-MGL [34], LSR [8], JCOD [65], C^2F-Net [30], SINetV2 [1], PreyNet [26], ERRNet [31], CubeNet [66], FBNet [67], DTC-Net [24], FindNet [68], FAP-Net [25], C^2F-NetV2 [30], BGNet [35], BSA-Net [36], SegMaR [33], R-MGL_V2 [69], PUENet [20], MSCAF-Net [32], and MRR-Net [28]. For fair comparison, all prediction maps of these methods are either obtained from their websites or generated by their source codes released by the authors. In addition, the prediction maps are evaluated with the same codes under the same evaluation protocols. As delineated in [32] and [44], the COD task exhibits sensitivity to the resolution of input images. Since PUENet [20] defaults to using a 512 × 512 input resolution, this is much larger than our setting. For fair comparison, we retrain PUENet [20], modifying only the input resolution to 384 × 384 based on the source code released by the authors.

1) Quantitative Comparison: The quantitative comparison results with SOTA methods are listed in Table I. We report the mean and the standard deviation of the metrics over three independent trials. On the one hand, our proposed method achieves high performance on large-scale datasets, e.g., COD10K and NC4K. In particular, although MSCAF-Net [32] achieves the second-best performance on NC4K, S_α, E_ϕ, and F_β^w of our SAE-Net are increased by 3.0%, 2.5%, and 6.0% on COD10K, respectively. Considering both COD10K and NC4K, MRR-Net [28] achieves the second-best performance. On CAMO, our model outperforms MRR-Net [28] by 1.1%, 1.1%, 1.1%, 0.6% in terms of S_α, E_ϕ, F_β^w, and M, respectively. On the other hand, our SAE-Net consistently achieves competitive performance across four datasets. SegMaR [33] achieves the best performance on CHAMELEON. MRR-Net [28] both achieves the second-best performance on CAMO and COD10K. MSCAF-Net [32] achieves the second-best performance on NC4K. For each model, our method outperforms its results on the other three datasets. The above analysis demonstrates the superior performance of our method.

In addition, it is noticeable that certain methods achieving high performance on CHAMELEON dataset may demonstrate inferior performance on other large-scale datasets. For instance, SegMaR [33] achieves the best performance on the CHAMELEON dataset. Comparatively, on the largest test set NC4K, our SAE-Net enhances S_α, E_ϕ, and F_β^w by 2.1%, 1.6%, and 1.5%, respectively, while decreasing M by 0.4%. This variation may be attributed to the testing biases in small-scale dataset as mentioned in [83].

2) Qualitative Comparison: The qualitative comparison results with SOTA methods are shown in Fig. 7. The samples contain several challenging scenarios including complex structure (rows 1 and 2), multiple objects (rows 1–4), small objects (rows 5 and 6), high similarity (rows 7 and 8), and occlusion (rows 9 and 10). For all the challenging scenarios, our model obtains superior predictions, while the comparison results usually have omission areas (e.g., rows 2, 3, and 8) or false alarms (e.g., rows 1, 6, and 7). This can be attributed to the capability of our SAE-Net to aggregate sufficient multilevel semantics and explore detailed multiscale features at each decoding scale. This enables the model to fully leverage contextual semantics for detecting camouflaged objects of different locations and scales. In addition, SAE-Net captures edge semantics with an explicit boundary knowledge guidance, aiding in more precise predictions of object edge shapes.

Fig. 7. Qualitative comparison with SOTA methods. Our model achieves superior visual performance in challenging scenarios.

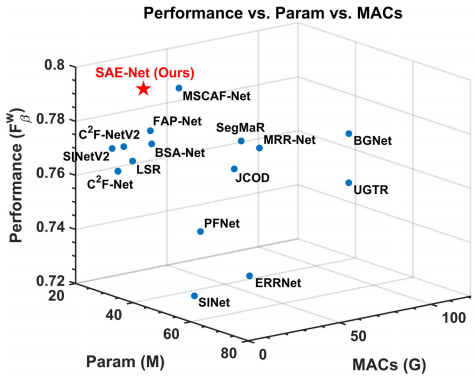


Fig. 8. Performance, parameters, and MACs compared with SOTA methods. The performance is measured using the weighted F-measure ($\mathbf{F}_\beta^w$) [78] on the NC4K dataset [8]. Our SAE-Net achieves superior performance with competitive model complexity.

TABLE II
BDE RESULTS OF SOTA METHODS

Method	CHAMELEON (76)	CAMO (250)	COD10K (2,026)	NC4K (4,121)
R-MGL [34]	15.26	20.64	27.02	17.43
BSA-Net [36]	13.38	17.51	24.21	15.63
FAP-Net [25]	15.01	18.24	27.35	17.32
BGNet [35]	15.22	16.43	23.57	15.66
SAE-Net (Ours)	**12.38**	**15.44**	**21.60**	**13.91**

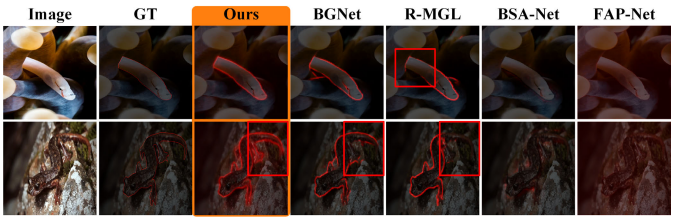


Fig. 9. Boundary visual comparison with SOTA methods. Our model generates more complete and detailed boundaries.

3) Model Complexity Comparison: To evaluate the model complexity of our SAE-Net, we compare it with 15 open-source models using a unified test code, including model parameters and multiply–accumulate operations (MACs). As shown in Fig. 8, our SAE-Net achieves superior performance on the largest NC4K test set and is more lightweight than the top-performing MSCAF-Net [32], MRR-Net [28], SegMaR [33], and BGNet [35]. Compared with BSA-Net [36],

TABLE III
COMPARISON WITH TRANSFORMER-BASED METHODS. † DENOTES THE RESULTS ARE DIRECTLY CITED FROM THE ORIGINAL PAPERS. † DENOTES THE RESULTS ARE UNAVAILABLE

Method	Publication	Backbone	CHAMELEON (76)				CAMO (250)				COD10K (2,026)				NC4K (4,121)			
			$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$M \downarrow$
DITNet [83]	ICPR'22	SegFormer [16]	0.883	0.929	0.813	0.033	0.856	0.916	0.796	0.050	0.824	0.896	0.695	0.034	0.863	0.917	0.792	0.041
FSPNet [80]	CVPR'23	ViT [84]	0.908	0.943	0.851	0.023	0.856	0.899	0.799	0.050	0.851	0.895	0.735	0.026	0.879	0.915	0.816	0.035
OPNet† [85]	IJCV'23	Conformer [86]	0.908	-	0.866	0.022	0.858	-	0.817	0.050	0.857	-	0.767	0.026	0.883	-	0.838	0.034
UEDG [87]	TMM'23	PVT [88]	0.911	0.958	**0.866**	0.023	0.863	0.922	0.817	0.048	0.858	0.924	0.766	0.025	0.879	0.929	0.830	0.035
JCNet† [81]	TIM'23	SwinT [89]	0.906	-	0.862	**0.021**	0.850	-	0.800	0.054	0.852	-	0.763	0.026	0.876	-	0.827	0.035
LSR+† [90]	TCSVT'23	SwinT [89]	0.895	0.943	-	0.025	0.854	0.924	-	0.049	0.847	0.924	-	0.028	0.870	0.924	-	0.036
PUENet [20]	TIP'23	Hybrid-ViT [84]	0.898	0.951	0.838	0.025	0.856	0.915	0.807	0.050	0.845	0.918	0.744	0.027	0.879	0.931	0.828	0.034
MSCAF-Net [32]	TCSVT'23	PVTv2 [91]	**0.912**	**0.958**	0.865	0.022	0.873	0.929	0.828	0.046	0.865	0.927	0.775	0.024	0.887	0.934	0.838	0.032
SAE-Net (Ours)	-	PVTv2 [91]	0.907 (±0.0007)	0.954 (±0.0027)	0.853 (±0.0032)	0.025 (±0.0007)	**0.880** (±0.0003)	**0.930** (±0.0011)	**0.834** (±0.0015)	**0.044** (±0.0005)	**0.868** (±0.0007)	**0.930** (±0.0008)	**0.778** (±0.0004)	**0.024** (±0.0004)	**0.890** (±0.0016)	**0.935** (±0.0012)	**0.840** (±0.0022)	**0.032** (±0.0007)

which has a comparable model complexity, our model outperforms it on NC4K by 2.5% in terms of F_β^w. The above analysis indicates the efficiency of our method.

4) Boundary Comparison: We compare our SAE-Net with representative boundary-related methods, i.e., BGNet [35], R-MGL [34], BSA-Net [36], and FAP-Net [25]. For a fair comparison, the boundary maps of these methods are generated by their source codes released by the authors. Following [92], [93], the boundary displacement error (BDE) is adopted to evaluate the precision of boundaries. The smaller the BDE, the better the result. The quantitative results are listed in Table II. It can be observed that our SAE-Net achieves superior performance. As shown in Fig. 9, we visualize the boundaries. It can be observed that BSA-Net [36] roughly locates coarse object contours, while FAP-Net [25] acts more like a general edge extractor. Compared with BGNet [35] and R-MGL [34], our model generates more intact and continuous edges.

D. Adaptability to Transformer Architecture

Vision transformer [84] has shown strong capability in modeling long-range dependencies and exhibited remarkable aptitude in COD task [64], [90]. To validate the effectiveness of our key components for the transformer architecture, we utilize the improved pyramid vision transformer (PVTv2) [91] as backbone, which is also adopted in [32] and [87]. We compare our model with eight transformer-based SOTA methods, i.e., DITNet [83], LSR+ [90], PUENet [20], MSCAF-Net [32], FSPNet [80], OPNet [85], JCNet [81], and UEDG [87].

The experimental results are listed in Table III. We report the mean and the standard deviation of the metrics over three independent trials. It can be observed that our model achieves comparable performance compared with SOTA methods, which indicates the generality of our proposed framework for the transformer architecture. Visual comparisons are shown in Fig. 10. Several challenging scenarios are included, i.e., complex structure (row 1), high similarity with background (row 2), multiple objects (row 3), small objects (row 4), occlusion (row 5), and low illumination (row 6). It can be observed that our model achieves superior performance in the above challenging scenarios.

E. Ablation Study

To comparatively discuss the effects of the different components in our SAE-Net, we perform ablation analysis on COD10K [2] and NC4K [8] datasets.

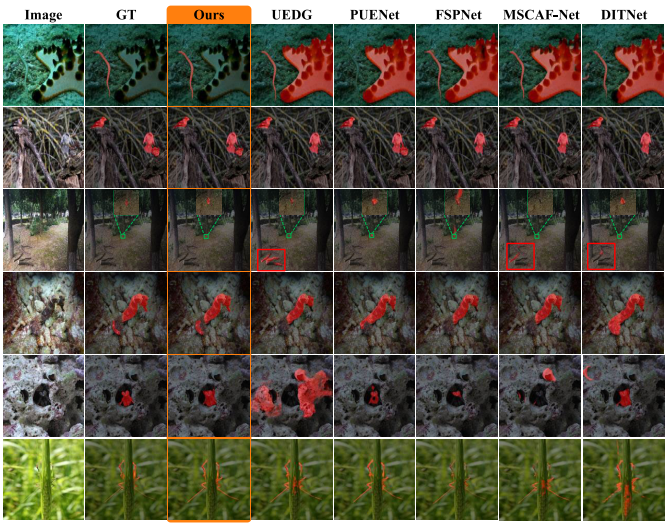


Fig. 10. Visual comparison with SOTA transformer-based methods.

TABLE IV
EFFECTS OF THE MAIN COMPONENTS

Model	WDSA	DEMS	EBM	COD10K (2,026)				NC4K (4,121)			
				$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$M \downarrow$
(a)		✓	✓	0.833	0.901	0.721	0.032	0.857	0.907	0.792	0.044
(b)	✓		✓	0.834	0.901	0.717	0.032	0.859	0.909	0.790	0.043
(c)	✓	✓		0.829	0.891	0.698	0.034	0.857	0.903	0.776	0.046
(d)	✓	✓	✓	**0.838**	**0.904**	**0.727**	**0.031**	**0.862**	**0.912**	**0.796**	**0.042**

1) Effects of the Main Components: Our model consists of three key parts, i.e., WDSA, DEMS, and EBM. We conduct ablation studies on the three main components, and the results are shown in Table IV. We remove each of the three key parts from the final model (d), and get (a), (b), and (c), respectively. It can be observed that the performance will decrease with the removal of any module, indicating the effectiveness of the three components.

2) Effects of the WDSA: The ablation study results of WDSA are shown in (a)–(g) of Table V. (a) and (b) denote removing ILS and ILF from WDSA, respectively. (c) and (d) represent replacing SE [59] blocks in ILS with the widely used bottleneck attention module (BAM) [94] and convolutional block attention module (CBAM) [95], respectively. (e) denotes replacing our WDSA with a dense connection, which is a common feature aggregation strategy. To explore the necessity of fusing all level features at each scale, we conduct the following experiments. (f) represents replacing our WDSA with adjacent two-level fusion method, namely cross-level feature fusion (CLFF), proposed in [25]. (g) denotes using the three-level fusion in WDSA instead of the five-level fusion.

TABLE V
ABLATION STUDIES OF THE WDSA, DEMS, AND EBM

Model	COD10K (2,026)				NC4K (4,121)			
	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$M \downarrow$
(a) w/o ILS	0.835	0.900	0.716	0.032	0.859	0.908	0.788	0.044
(b) w/o ILF	0.833	0.898	0.715	0.033	0.859	0.909	0.789	0.044
(c) SE [59]→BAM [94]	0.833	0.902	0.715	0.032	0.860	0.909	0.789	0.043
(d) SE [59]→CBAM [95]	0.834	0.900	0.721	0.031	0.859	0.909	0.792	0.043
(e) WDSA→DenseConn	0.833	0.897	0.712	0.033	0.861	0.910	0.790	0.043
(f) WDSA→CLFF [25]	0.834	0.901	0.720	0.031	0.858	0.906	0.789	0.044
(g) Three-level Fusion	0.834	0.903	0.721	0.032	0.857	0.908	0.789	0.044
(h) w/o DConv	0.831	0.899	0.708	0.033	0.858	0.906	0.784	0.045
(i) w/o AConv	0.835	0.902	0.718	0.032	0.861	0.910	0.791	0.043
(j) DEMS→ASPP [63]	0.831	0.895	0.707	0.034	0.858	0.907	0.786	0.045
(k) DEMS→ERF [32]	0.829	0.895	0.703	0.034	0.860	0.908	0.786	0.044
(l) DEMS→RFB [1]	0.833	0.901	0.714	0.032	0.858	0.909	0.788	0.044
(m) w/o EKGBL	0.834	0.901	0.716	0.032	0.858	0.907	0.787	0.044
(n) w/o GAF	0.825	0.887	0.698	0.035	0.854	0.902	0.778	0.046
(o) EBM→MECSA [39]	0.833	0.902	0.714	0.032	0.857	0.908	0.786	0.044
(p) EBM→SEA [36]	0.833	0.900	0.709	0.032	0.860	0.907	0.787	0.044
(q) Ours	**0.838**	**0.904**	**0.727**	**0.031**	**0.862**	**0.912**	**0.796**	**0.042**

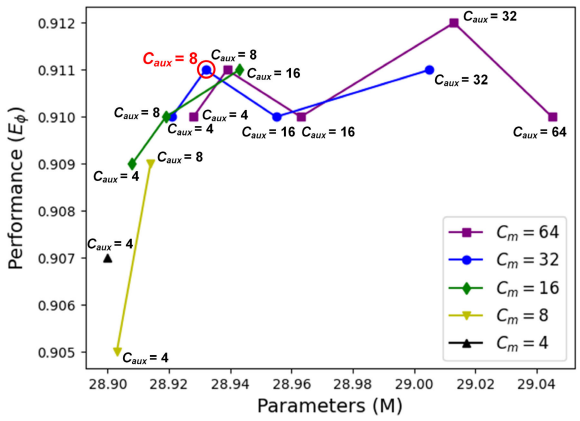


Fig. 11. Performance and parameter comparison of different C_m and C_{aux}. The performance is measured using the E-measure (E_ϕ) on the NC4K dataset [8]. The model circled in red is our final model.

Seeing the results of (a) and (b), removing ILS or ILF in WDSA causes a performance degradation compared with (q), indicating the effectiveness of both ILS and ILF. The result of (q) is slightly higher than (c) and (d), so we choose SE block in ILS. Compared with (e), (f), and (g), our WDSA module outperforms the other multilevel fusion strategies. This demonstrates the effectiveness of our WDSA strategy. The performance of (f), (g), and (q) gradually increases, indicating the effects of integrating more level features at each decoding scale.

3) Chosen of C_m and C_{aux} in WDSA: Directly fusing low- and high-level features could lead to redundancy. Hence, selecting prominent information from multilevel features is crucial for ensuring efficiency. In Fig. 11, we compare performance and model parameters across different C_m and C_{aux}. Here, we test $C_m \in \{64, 32, 16, 8, 4\}$ and vary C_{aux} from 4 to C_m for each C_m. Observations indicate a declining performance trend as C_{aux} decreases for each C_m. When $C_m > 16$, the decline is relatively gradual; however, for $C_m \leq 16$, the performance drops significantly. This suggests redundancy in numerous feature channels, but excessive information loss leads to a notable performance degradation. Considering both performance and model parameters, we choose $C_m = 32$ and $C_{\text{aux}} = 8$ as our final model, which balances between the richness of multilevel features and parameter efficiency.

Fig. 12. Samples that are selected based on the weights assigned by the WDSA modules. These samples are taken from the test set of the COD10K [2] dataset. "5-4-3" means that WDSA modules assign the three largest weights to the fifth-, fourth-, and third-level features. We can observe that our model more emphasizes on high-level features in images with higher visual camouflage levels.

4) Exploration of Multilevel Weights in WDSA: To explore the relationship between multilevel features and camouflaged objects, we visualize the weights of WDSA modules. To obtain the weights of the five-level features assigned by our model, we average the learned weights of the five WDSA modules for each input image. We obtain the weights of the images in the test set of the COD10K [2] dataset and find that the model's dependence on different level features is related to the visual camouflage level of the input image. We present some samples in Fig. 12, where "5-4-3" denotes that the top three weights are assigned to the fifth-, fourth-, and third-level features. We can observe that our model pays more attention to high-level features in images with higher visual camouflage levels. This observation might be explained as follows. Intuitively, objects with lower camouflage levels typically exhibit more noticeable texture and color disparities compared to their surrounding environment. As low-level features encapsulate rich texture and color characteristics, they may be more effective in detecting objects with lower camouflage levels. Conversely, objects with higher camouflage levels often lack clear, salient features, making them more difficult to detect. The subtle traces that render camouflaged objects detectable typically manifest in specific regions. Uncovering these nuances may require higher-level features with a broader contextual understanding.

To further investigate the above observation, we conduct experiments on the CAM-FR [8] dataset. It consists of 2280 images with camouflaged object ranking annotations and encompasses three levels of camouflage, i.e., hard, median, and easy. For each image of a specific camouflage level, we count the occurrence of the most attended features and apply *softmax* operation to the statistical results. The results, as shown in Fig. 13, represent the distribution of the network's average dependence on the five-level features when detecting samples of different camouflage levels in the CAM-FR [8] dataset. Our results indicate that, for images with hard camouflage, the WDSA modules assign large weights to high-level features such as the fourth and fifth levels, suggesting that our model relies more heavily on these features to detect highly camouflaged objects. In contrast, for images with easy camouflage,

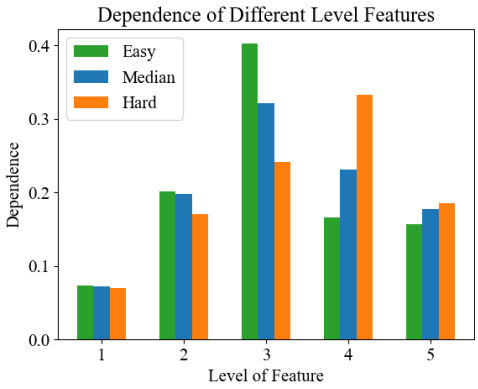


Fig. 13. Visualizations of the weights assigned by the WDSA modules to the five-level features on the CAM-FR [8] dataset, which consists of images with three camouflage levels. It shows the average dependence of the five-level features for detecting images with different camouflage levels. It can be observed that our method focuses on high-, middle-, and low-level features for the images of hard, median, and easy camouflage, respectively.

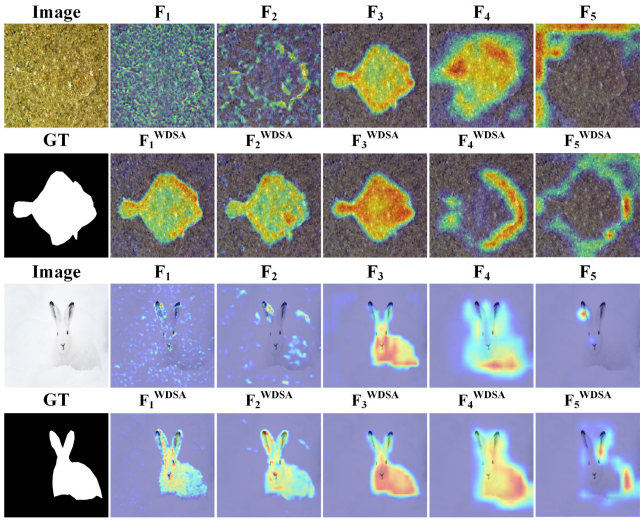


Fig. 14. Visualizations of the WDSA module. $\{\mathbf{F}_i\}_{i=1}^{5}$ denotes the five-level backbone features. $\{\mathbf{F}_i^{\text{WDSA}}\}_{i=1}^{5}$ indicates the outputs of the five WDSA modules.

our model emphasizes on low-level features. While for images with median camouflage, the third-level feature is assigned the largest weight by the WDSA modules, and all five weights fall between those assigned for images with hard and easy camouflage.

The above observations and analysis demonstrate that our WDSA modules can adaptively fuse the multilevel features according to different scenes. This aligns well with our initial design intentions. Also, our visualization results reveal the correlation between multilevel features and different camouflage levels. This study might offer valuable insights for addressing the camouflaged object ranking task closely associated with camouflage levels, as well as designing multilevel aggregation structures tailored for COD models in specific scenarios.

5) Visualizations of the WDSA: To illustrate the adaptive aggregation of multilevel semantics by our WDSA module, we visualize the corresponding features in Fig. 14. It can be observed that low-level backbone features (e.g., $\mathbf{F}_1$) contain rich texture cues but lack camouflage semantics, whereas high-level backbone features (e.g., $\mathbf{F}_5$) contain coarse

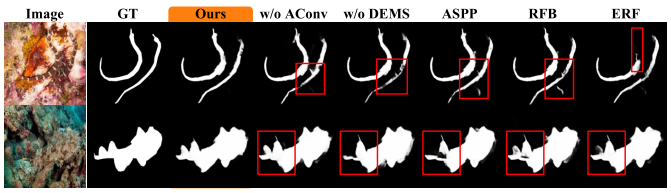


Fig. 15. Visualizations of the DEMS module. The third–fifth columns show the detection results of our model, our model without AConv, and our model without DEMS, respectively. The sixth to eighth columns show the results of replacing DEMS with ASPP [63], RFB [1], and ERF [32].

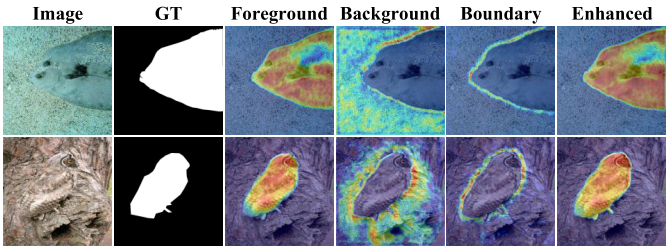


Fig. 16. Feature visualizations of the EBM. The third to sixth columns show the visualizations of the foreground ($\mathbf{F}_i^f$), background ($\mathbf{F}_i^b$), boundary ($\mathbf{F}_i^e$), and enhanced features ($\mathbf{F}_i^{\text{EBM}}$) of the EBM.

location information but lack object details. After aggregating multilevel semantics through the WDSA module, low-level features can focus more on camouflaged object regions, while high-level features can utilize texture and shape cues to achieve more accurate location. This demonstrates that our WDSA module can adaptively aggregate multilevel semantics at different decoding scales, thereby sufficiently enhancing the original backbone features.

6) Effects of the DEMS: The ablation study results of DEMS are shown in (h)–(l) of Table V. We provide five different variants: (h) removing the DConv branch in the DeConv; (i) removing the AConv branch in the DeConv; (j), (k), and (l) denote replacing the DEMS with the ASPP [63], ERF [32], and RFB [1], respectively. Comparing the results of (h), (i), and (q), removing either DConv or AConv in the DeConv leads to performance degradation, and removing DConv leads to a more substantial performance decline. It suggests that multiscale contexts are useful for COD, and AConv can complement DConv by providing detailed clues to enhance COD performance. Seeing the results of (j), (k), (l), our DEMS module outperforms existing multiscale modules, i.e., ASPP [63], ERF [32], and RFB [1]. This is primarily attributed to the enhancement of local detailed information. The ASPP and RFB use multiple DConvs with different dilation rates, but lack detailed information that may contain subtle camouflaged clues. The ERF module only adopts a dilation rate of 3, while our DEMS module adopts multiple dilation rates and interacts contexts between multiscale branches to obtain richer multiscale features.

7) Visualizations of the DEMS: Visualizations of the DEMS module are shown in Fig. 15. Comparing the results of our model and "w/o AConv," our model generates finer object structures, indicating that AConv can complement diluted multiscale features with subtle details. Compared with ASPP [63], RFB [1], and ERF [32], our predictions contain clearer detailed structures as shown in the red boxes. This indicates that our

TABLE VI
QUANTITATIVE COMPARISON WITH SOD METHODS

Method	Publication	DUTS-TE [96]			DUT-OMRON [97]			ECSSD [98]			HKU-IS [99]		
		$F_\beta \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$M \downarrow$	$S_\alpha \uparrow$
LDF [100]	CVPR'20	0.855	0.034	0.892	0.773	0.051	0.839	0.930	0.034	0.925	0.914	0.027	0.920
DCNet [101]	TIP'21	0.877	0.034	0.892	0.782	0.052	0.839	0.938	0.034	0.925	0.929	0.028	0.920
PurNet [102]	TIP'21	0.859	0.039	0.869	0.782	**0.051**	0.841	0.936	0.035	0.925	0.926	0.031	0.918
PDRNet [103]	TCSVT'22	0.876	0.035	0.877	0.795	0.052	0.846	0.941	0.032	0.927	0.933	0.027	0.924
DNA [104]	TCYB'22	0.874	0.047	0.858	0.803	0.063	0.818	0.940	0.043	0.915	0.928	0.036	0.905
TCRNet [105]	TCSVT'23	0.878	0.034	0.880	0.791	0.054	0.843	0.943	0.031	0.928	0.933	0.026	0.923
PoolNet+ [22]	TPAMI'23	0.866	0.043	0.875	0.791	0.060	0.829	0.939	0.045	0.915	0.925	0.037	0.908
ICON-R [106]	TPAMI'23	0.876	0.037	0.889	0.799	0.057	0.845	0.943	0.032	0.929	0.931	0.029	0.920
SAE-Net (Ours)	-	**0.889**	**0.034**	**0.900**	**0.803**	0.057	**0.847**	**0.950**	**0.028**	**0.936**	**0.938**	**0.025**	**0.928**

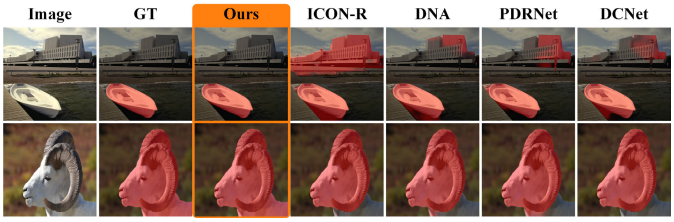


Fig. 17. Qualitative comparison with SOD methods.

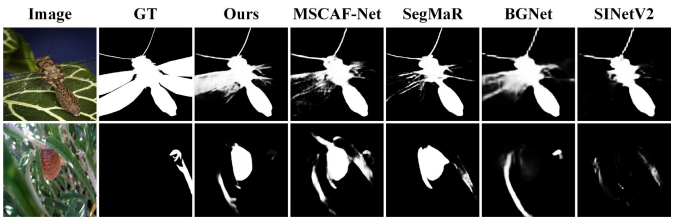


Fig. 18. Some failure cases of our model and SOTA methods. The extremely challenging scenes include objects with transparent regions (row 1) and salient distraction (row 2).

DEMS module can capture finer details than these multiscale modules.

8) Effects of the EBM: The ablation study results of EBM are shown in (m)–(p) of Table V. To evaluate the effects of the EKGBL and GAF modules in EBM, we provide two variants: (m) EBM without the EKGBL and (n) EBM without the GAF modules. Also, we replace our EBM with existing boundary learning and fusion modules, i.e., MECSA [39] and SEA [36], and get the model (o) and (p), respectively. Comparing the results of (m) and (q), removing EKGBL causes a significant drop of 1.1% and 0.9% in terms of the F_β^w on COD10K and NC4K, respectively. This demonstrates that generating boundary semantics through explicit boundary knowledge guidance is beneficial for COD. Seeing the results of (n), removing GAF modules leads to a decrease in performance, indicating the effectiveness of the GAF module. It can be observed that our model outperforms (o) and (p), indicating the effectiveness of our EBM module.

9) Visualizations of the EBM: To visually showcase the functions of the EBM module, we conduct feature visualizations as shown in Fig. 16. It can be observed that the foreground (col. 3) and background (col. 4) features focus on different but complementary regions when the boundary (col. 5) is accurately detected. This indicates that the EKGBL can distinguish the foreground and background features, and further obtain accurate boundary features from them. The restoration of omission areas (row 1) and more highlighted boundaries (row 2) in the enhanced features (col. 6) show more accurate detection results compared to the original foreground feature (col. 3). This indicates that the proposed GAF module can effectively fuse image and edge features.

V. APPLICATION TO SOD

SOD aims to detect the most visually distinctive objects from a given image and has gained great attention from computer vision community. In this experiment, we show the effectiveness of our proposed method in SOD. Following [22], three widely used metrics are adopted, i.e., the maximum F-measure (F_β), MAE (M), and the structure-measure (S_α). We train our SAE-Net with Res2Net-50 [53] backbone on DUTS-TR [96], and test on DUTS-TE [96], DUT-OMRON [97], ECSSD [98], and HKU-IS [99]. We compare our SAE-Net with eight SOTA SOD methods, i.e., LDF [100], DCNet [101], PurNet [102], PDRNet [103], DNA [104], TCRNet [105], PoolNet+ [22], and ICON-R [106]. The quantitative and qualitative comparison results are shown in Table VI and Fig. 17, respectively. It can be seen that our SAE-Net achieves competitive performance against SOTA SOD methods. This demonstrates that our proposed method can be well generalized to the SOD task.

VI. LIMITATIONS AND FUTURE WORK

We present some failure cases of our SAE-Net and SOTA methods in Fig. 18. These cases are mainly caused by two extreme challenging scenes, i.e., objects with transparent regions (row 1) and salient distraction (row 2). In the first row, it is difficult to completely segment the transparent wings of the dragonfly. The transparent nature of transparent parts causes them to blend with the background, while the appearance of their internal regions varies with different scenes. This makes them hard to detect by methods relying on RGB modality. In the second row, there are salient distracting objects in the background, and existing models tend to segment the salient objects. Since current COD methods are typically trained with camouflaged annotations, it may be difficult for them to deeply understand the essential differences between saliency and camouflage. The presence of both camouflaged and salient objects in the scene may lead to incorrect detection.

To address the limitations, future work can focus on the following two aspects. First, the multimodality learning might be helpful to detect transparent objects and parts. For example, the polarization cues contain intrinsic information about objects with different material properties. It has shown effectiveness for detecting transparent glasses [107]. Second, the joint training of camouflaged and salient data might help the model deeper understand the content of an image, facilitating both COD and SOD tasks. Therefore, in future work, we will further improve our method by introducing multimodality learning and the joint learning of both camouflaged and salient data.

VII. Conclusion

In this article, we propose a weighted dense SAE-Net for COD. Specifically, the WDSA module adopts ILS and ILF steps to sufficiently aggregate multilevel semantics at each decoding scale. The DEMS module further exploits refined multiscale features. The EBM module incorporates explicit boundary knowledge guidance into boundary learning and enhances the feature representation with edge cues. Extensive experiments demonstrate that our model achieves competitive performance against SOTA methods on both CNN- and transformer-based architectures. Simultaneously, the analysis of WDSA modules' weights suggests that the network relies more on higher-level features to detect camouflaged objects with higher camouflage levels. Visualizations indicate the superior performance of our model on various challenging scenarios. Ablation studies further demonstrate the effectiveness of the main components. In the future, we plan to introduce multimodality learning and the joint learning of both camouflaged and salient data to address the failure cases.

References

[1] D.-P. Fan, G.-P. Ji, M.-M. Cheng, and L. Shao, "Concealed object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, pp. 6024–6042, Oct. 2022.

[2] D.-P. Fan, G.-P. Ji, G. Sun, M.-M. Cheng, J. Shen, and L. Shao, "Camouflaged object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2774–2784.

[3] Z. Huang, C. Lv, Y. Xing, and J. Wu, "Multi-modal sensor fusion-based deep neural network for end-to-end autonomous driving with scene understanding," *IEEE Sensors J.*, vol. 21, no. 10, pp. 11781–11790, May 2021.

[4] D.-P. Fan et al., "Inf-Net: Automatic COVID-19 lung infection segmentation from CT images," *IEEE Trans. Med. Imag.*, vol. 39, no. 8, pp. 2626–2637, Aug. 2020.

[5] X. Wang et al., "SK-UNet: An improved U-Net model with selective kernel for the segmentation of LGE cardiac MR images," *IEEE Sensors J.*, vol. 21, no. 10, pp. 11643–11653, May 2021.

[6] S. Tian et al., "CASDD: Automatic surface defect detection using a complementary adversarial network," *IEEE Sensors J.*, vol. 22, no. 20, pp. 19583–19595, Oct. 2022.

[7] W. Ouyang and Y. Wei, "An anchor-free detector with channel-based prior and bottom-enhancement for underwater object detection," *IEEE Sensors J.*, vol. 23, no. 20, pp. 24800–24811, Oct. 2023.

[8] Y. Lv et al., "Simultaneously localize, segment and rank the camouflaged objects," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 11586–11596.

[9] X. Cheng et al., "Implicit motion handling for video camouflaged object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 13864–13873.

[10] Z. Wu et al., "Source-free depth for object pop-out," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 1032–1042.

[11] Q. Wang, J. Yang, X. Yu, F. Wang, P. Chen, and F. Zheng, "Depth-aided camouflaged object detection," in *Proc. 31st ACM Int. Conf. Multimedia*, Oct. 2023, pp. 3297–3306.

[12] P. Sun, T. Liu, X. Chen, S. Zhang, Y. Zhao, and S. Wei, "Multi-source aggregation transformer for concealed object detection in millimeter-wave images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 9, pp. 6148–6159, Sep. 2022.

[13] X. Wang et al., "Self-Paced feature attention fusion network for concealed object detection in millimeter-wave image," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 1, pp. 224–239, Jan. 2022.

[14] H. Bi, C. Zhang, K. Wang, J. Tong, and F. Zheng, "Rethinking camouflaged object detection: Models and datasets," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 9, pp. 5708–5724, Sep. 2022.

[15] T. Y. Lin, P. Dollàr, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2117–2125.

[16] E. Xie et al., "SegFormer: Simple and efficient design for semantic segmentation with transformers," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 34, Dec. 2021, pp. 12077–12090.

[17] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[18] B. Yin, X. Zhang, Q. Hou, B.-Y. Sun, D.-P. Fan, and L. Van Gool, "CamoFormer: Masked separable attention for camouflaged object detection," 2022, *arXiv:2212.06570*.

[19] C. He et al., "Camouflaged object detection with feature decomposition and edge reconstruction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 22046–22055.

[20] Y. Zhang, J. Zhang, W. Hamidouche, and O. Deforges, "Predictive uncertainty estimation for camouflaged object detection," *IEEE Trans. Image Process.*, vol. 32, pp. 3580–3591, 2023.

[21] L. Liu et al., "Deep learning for generic object detection: A survey," *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 261–318, Feb. 2020.

[22] J.-J. Liu, Q. Hou, Z.-A. Liu, and M.-M. Cheng, "PoolNet+: Exploring the potential of pooling for salient object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 887–904, Jan. 2023.

[23] H. Mei, G. Ji, Z. Wei, X. Yang, X. Wei, and D. Fan, "Camouflaged object segmentation with distraction mining," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8768–8777.

[24] W. Zhai, Y. Cao, H. Xie, and Z.-J. Zha, "Deep Texton-coherence network for camouflaged object detection," *IEEE Trans. Multimedia*, vol. 25, pp. 5155–5165, 2023, doi: 10.1109/TMM.2022.3188401.

[25] T. Zhou, Y. Zhou, C. Gong, J. Yang, and Y. Zhang, "Feature aggregation and propagation network for camouflaged object detection," *IEEE Trans. Image Process.*, vol. 31, pp. 7036–7047, 2022.

[26] M. Zhang, S. Xu, Y. Piao, D. Shi, S. Lin, and H. Lu, "PreyNet: Preying on camouflaged objects," in *Proc. 30th ACM Int. Conf. Multimedia*, Oct. 2022, pp. 5323–5332.

[27] G. Chen, S.-J. Liu, Y.-J. Sun, G.-P. Ji, Y.-F. Wu, and T. Zhou, "Camouflaged object detection via context-aware cross-level fusion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 10, pp. 6981–6993, Oct. 2022.

[28] X. Yan, M. Sun, Y. Han, and Z. Wang, "Camouflaged object segmentation based on matching–recognition–refinement network," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–15, 2023, doi: 10.1109/TNNLS.2023.3291595.

[29] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269.

[30] Y. Sun, G. Chen, T. Zhou, Y. Zhang, and N. Liu, "Context-aware cross-level fusion network for camouflaged object detection," in *Proc. 30th Int. Joint Conf. Artif. Intell.*, Aug. 2021, pp. 1025–1031.

[31] G.-P. Ji, L. Zhu, M. Zhuge, and K. Fu, "Fast camouflaged object detection via edge-based reversible re-calibration network," *Pattern Recognit.*, vol. 123, Mar. 2022, Art. no. 108414.

[32] Y. Liu, H. Li, J. Cheng, and X. Chen, "MSCAF-Net: A general framework for camouflaged object detection via learning multi-scale context-aware features," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 9, pp. 4934–4947, 2023, doi: 10.1109/TCSVT.2023.3245883.

[33] Q. Jia, S. Yao, Y. Liu, X. Fan, R. Liu, and Z. Luo, "Segment, magnify and reiterate: Detecting camouflaged objects the hard way," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 4703–4712.

[34] Q. Zhai, X. Li, F. Yang, C. Chen, H. Cheng, and D.-P. Fan, "Mutual graph learning for camouflaged object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 12992–13002.

[35] Y. Sun, S. Wang, C. Chen, and T. Xiang, "Boundary-guided camouflaged object detection," in *Proc. 21st Int. Joint Conf. Artif. Intell.*, 2022, pp. 1335–1341.

[36] H. Zhu et al., "I can find you! Boundary-guided separated attention network for camouflaged object detection," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2022, pp. 3608–3616.

[37] J. Zhao, J.-J. Liu, D.-P. Fan, Y. Cao, J. Yang, and M.-M. Cheng, "EGNet: Edge guidance network for salient object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8778–8787.

[38] W. Wang, S. Zhao, J. Shen, S. C. H. Hoi, and A. Borji, "Salient object detection with pyramid attention and salient edges," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1448–1457.

[39] J. Wu, W. Liang, F. Hao, and J. Xu, "Mask-and-edge co-guided separable network for camouflaged object detection," *IEEE Signal Process. Lett.*, vol. 30, pp. 748–752, 2023.

[40] Y. Pang, X. Zhao, T.-Z. Xiang, L. Zhang, and H. Lu, "Zoom in and out: A mixed-scale triplet network for camouflaged object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 2150–2160.

[41] Y. Zhong, B. Li, L. Tang, S. Kuang, S. Wu, and S. Ding, "Detecting camouflaged object in frequency domain," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 4494–4503.

[42] X. Hu et al., "High-resolution iterative feedback network for camouflaged object detection," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2023, vol. 37, no. 1, pp. 881–889.

[43] Z. Song, X. Kang, X. Wei, H. Liu, R. Dian, and S. Li, "FSNet: Focus scanning network for camouflaged object detection," *IEEE Trans. Image Process.*, vol. 32, pp. 2267–2278, 2023.

[44] H. Xing, Y. Wang, X. Wei, H. Tang, S. Gao, and W. Zhang, "Go closer to see better: Camouflaged object detection via object area amplification and figure-ground conversion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 10, pp. 5444–5457, 2023, doi: 10.1109/TCSVT.2023.3255304.

[45] Y. Liu, Y. Duan, and T. Zeng, "Learning multi-level structural information for small organ segmentation," *Signal Process.*, vol. 193, Apr. 2022, Art. no. 108418.

[46] J. Wang, W. Li, Y. Gao, M. Zhang, R. Tao, and Q. Du, "Hyperspectral and SAR image classification via multiscale interactive fusion network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 12, pp. 10823–10837, Jun. 2023.

[47] J. Wang, W. Li, M. Zhang, R. Tao, and J. Chanussot, "Remote-sensing scene classification via multistage self-guided separation network," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–12, 2023, doi: 10.1109/TGRS.2023.3295797.

[48] M. Zhang, W. Li, X. Zhao, H. Liu, R. Tao, and Q. Du, "Morphological transformation and spatial-logical aggregation for tree species classification using hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5501212.

[49] J. Wang, M. Zhang, W. Li, and R. Tao, "A multistage information complementary fusion network based on flexible-mixup for HSI-X image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–13, 2023, doi: 10.1109/TNNLS.2023.3300903.

[50] J. Wang, W. Li, M. Zhang, and J. Chanussot, "Large kernel sparse convnet weighted by multi-frequency attention for remote sensing scene understanding," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–12, 2023, doi: 10.1109/TGRS.2023.3333401.

[51] F. Liu, Z. Hua, J. Li, and L. Fan, "MFBGR: Multi-scale feature boundary graph reasoning network for polyp segmentation," *Eng. Appl. Artif. Intell.*, vol. 123, Aug. 2023, Art. no. 106213.

[52] M. Zhen et al., "Joint semantic segmentation and boundary detection using iterative pyramid contexts," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13663–13672.

[53] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr, "Res2Net: A new multi-scale backbone architecture," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 2, pp. 652–662, Feb. 2021.

[54] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[55] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6230–6239.

[56] J.-J. Liu, Q. Hou, and M.-M. Cheng, "Dynamic feature integration for simultaneous detection of salient object, edge, and skeleton," *IEEE Trans. Image Process.*, vol. 29, pp. 8652–8667, 2020.

[57] J.-J. Liu, Q. Hou, M.-M. Cheng, J. Feng, and J. Jiang, "A simple pooling-based design for real-time salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3912–3921.

[58] T. Wang, A. Borji, L. Zhang, P. Zhang, and H. Lu, "A stagewise refinement model for detecting salient objects in images," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4039–4048.

[59] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2018, pp. 7132–7141.

[60] Y. Mao, Q. Jiang, R. Cong, W. Gao, F. Shao, and S. Kwong, "Cross-Modality fusion and progressive integration network for saliency prediction on stereoscopic 3D images," *IEEE Trans. Multimedia*, vol. 24, pp. 2435–2448, 2022.

[61] Y. Liu, K. Zhang, Y. Zhao, H. Chen, and Q. Liu, "Bi-RRNet: Bi-level recurrent refinement network for camouflaged object detection," *Pattern Recognit.*, vol. 139, Jul. 2023, Art. no. 109514.

[62] W. Liang, J. Wu, Y. Wu, X. Mu, and J. Xu, "FINet: Frequency injection network for lightweight camouflaged object detection," *IEEE Signal Process. Lett.*, vol. 31, pp. 526–530, 2024.

[63] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[64] F. Yang et al., "Uncertainty-guided transformer reasoning for camouflaged object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 4126–4135.

[65] A. Li, J. Zhang, Y. Lv, B. Liu, T. Zhang, and Y. Dai, "Uncertainty-aware joint salient object and camouflaged object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 10066–10076.

[66] M. Zhuge, X. Lu, Y. Guo, Z. Cai, and S. Chen, "CubeNet: X-shape connection for camouflaged object detection," *Pattern Recognit.*, vol. 127, Jul. 2022, Art. no. 108644.

[67] J. Lin, X. Tan, K. Xu, L. Ma, and R. W. H. Lau, "Frequency-aware camouflaged object detection," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 19, no. 2, pp. 1–16, Mar. 2023.

[68] P. Li, X. Yan, H. Zhu, M. Wei, X.-P. Zhang, and J. Qin, "FindNet: Can you find me? Boundary-and-texture enhancement network for camouflaged object detection," *IEEE Trans. Image Process.*, vol. 31, pp. 6396–6411, 2022.

[69] Q. Zhai et al., "MGL: Mutual graph learning for camouflaged object detection," *IEEE Trans. Image Process.*, vol. 32, pp. 1897–1910, 2023.

[70] H. Li, G. Chen, G. Li, and Y. Yu, "Motion guided attention for video salient object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7273–7282.

[71] J. Cao, C. Ma, T. Yao, S. Chen, S. Ding, and X. Yang, "End-to-end reconstruction-classification learning for face forgery detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 4103–4112.

[72] J. Wei, S. Wang, and Q. Huang, "F^3Net: Fusion, feedback and focus for salient object detection," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 7, 2020, pp. 12321–12328.

[73] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020.

[74] P. Skurowski, H. Abdulameer, J. Błaszczyk, T. Depta, A. Kornacki, and P. Kozieł, "Animal camouflage analysis: Chameleon database," *Unpublished Manuscript*, vol. 2, no. 6, p. 7, 2018.

[75] T.-N. Le, T. V. Nguyen, Z. Nie, M.-T. Tran, and A. Sugimoto, "Anabranch network for camouflaged object segmentation," *Comput. Vis. Image Understand.*, vol. 184, pp. 45–56, Jul. 2019.

[76] D.-P. Fan, M.-M. Cheng, Y. Liu, T. Li, and A. Borji, "Structure-measure: A new way to evaluate foreground maps," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4558–4567.

[77] D.-P. Fan, C. Gong, Y. Cao, B. Ren, M.-M. Cheng, and A. Borji, "Enhanced-alignment measure for binary foreground map evaluation," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 698–704.

[78] R. Margolin, L. Zelnik-Manor, and A. Tal, "How to evaluate foreground maps," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 248–255.

[79] F. Perazzi, P. Krähenbühl, Y. Pritch, and A. Hornung, "Saliency filters: Contrast based filtering for salient region detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 733–740.

[80] Z. Huang, H. Dai, S. Wang, T. Xiang, H. Chen, and J. Qin, "Feature shrinkage pyramid for camouflaged object detection with transformers," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 5557–5566.

[81] X. Jiang et al., "Camouflaged object segmentation based on joint salient object for contrastive learning," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–16, 2023.

[82] D. P. Kingma and J. Ba, "ADAM: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–15. [Online]. Available: https://iclr.cc/archive/www/doku.php%3Fid=iclr2015:accepted-main.html

[83] Z. Liu, Z. Zhang, Y. Tan, and W. Wu, "Boosting camouflaged object detection with dual-task interactive transformer," in *Proc. 26th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2022, pp. 140–146.

[84] A. Dosovitskiy, L. Beyer, and A. Kolesnikov, "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021, pp. 1–21. [Online]. Available: https://iclr.cc/virtual/2021/poster/3013

[85] H. Mei et al., "Camouflaged object segmentation with omni perception," *Int. J. Comput. Vis.*, vol. 131, no. 11, pp. 3019–3034, Nov. 2023.

[86] Z. Peng et al., "Conformer: Local features coupling global representations for visual recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 357–366.

[87] Y. Lyu, H. Zhang, Y. Li, H. Liu, Y. Yang, and D. Yuan, "UEDG: Uncertainty-edge dual guided camouflage object detection," *IEEE Trans. Multimedia*, vol. 26, pp. 4050–4060, 2024.

[88] W. Wang et al., "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 548–558.

[89] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9992–10002.

[90] Y. Lv, J. Zhang, Y. Dai, A. Li, N. Barnes, and D.-P. Fan, "Towards deeper understanding of camouflaged object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 7, pp. 3462–3476, 2023, doi: 10.1109/TCSVT.2023.3234578.

[91] W. Wang et al., "PVT v2: Improved baselines with pyramid vision transformer," *Comput. Vis. Media*, vol. 8, no. 3, pp. 415–424, Sep. 2022.

[92] Y. Zeng, P. Zhang, Z. Lin, J. Zhang, and H. Lu, "Towards high-resolution salient object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7233–7242.

[93] H. Tang, S. Chen, Y. Liu, S. Wang, Z. Chen, and X. Hu, "Boundary-aware dichotomous image segmentation," *Vis. Comput.*, vol. 2024, pp. 1–12, Feb. 2024.

[94] J. Park, S. Woo, J.-Y. Lee, and I. S. Kweon, "BAM: Bottleneck attention module," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2019, pp. 147–161. [Online]. Available: https://bmva-archive.org.uk/bmvc/2018/index.html

[95] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19. [Online]. Available: https://openaccess.thecvf.com/content_ECCV_2018/html/Sanghyun_Woo_Convolutional_Block_Attention_ECCV_2018_paper.html

[96] L. Wang et al., "Learning to detect salient objects with image-level supervision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3796–3805.

[97] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, "Saliency detection via graph-based manifold ranking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3166–3173.

[98] Q. Yan, L. Xu, J. Shi, and J. Jia, "Hierarchical saliency detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 1155–1162.

[99] G. Li and Y. Yu, "Visual saliency based on multiscale deep features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5455–5463.

[100] J. Wei, S. Wang, Z. Wu, C. Su, Q. Huang, and Q. Tian, "Label decoupling framework for salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13022–13031.

[101] Z. Wu, L. Su, and Q. Huang, "Decomposition and completion network for salient object detection," *IEEE Trans. Image Process.*, vol. 30, pp. 6226–6239, 2021.

[102] J. Li, J. Su, C. Xia, M. Ma, and Y. Tian, "Salient object detection with purificatory mechanism and structural similarity loss," *IEEE Trans. Image Process.*, vol. 30, pp. 6855–6868, 2021.

[103] L. Zhang, Q. Zhang, and R. Zhao, "Progressive dual-attention residual network for salient object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 9, pp. 5902–5915, Sep. 2022.

[104] Y. Liu, M.-M. Cheng, X.-Y. Zhang, G.-Y. Nie, and M. Wang, "DNA: Deeply supervised nonlinear aggregation for salient object detection," *IEEE Trans. Cybern.*, vol. 52, no. 7, pp. 6131–6142, Jul. 2022.

[105] Q. Zhang, R. Zhao, and L. Zhang, "TCRNet: A trifurcated cascaded refinement network for salient object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 1, pp. 298–311, Jan. 2023.

[106] M. Zhuge, D.-P. Fan, N. Liu, D. Zhang, D. Xu, and L. Shao, "Salient object detection via integrity learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 3738–3752, Mar. 2023.

[107] Y. Qiao et al., "Multi-view spectral polarization propagation for video glass segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 23218–23228.

Weiyun Liang (Student Member, IEEE) received the bachelor's degree from Xidian University, Xi'an, China, in 2021. He is currently pursuing the Ph.D. degree with the College of Artificial Intelligence from Nankai University, Tianjin, China.

His research interests include computer vision and machine learning.

Jiesheng Wu (Student Member, IEEE) is currently pursuing the Ph.D. degree with the College of Artificial Intelligence, Nankai University, Tianjin, China.

His research interests include computer vision and multimodal computing.

Xinyue Mu is currently pursuing the master's degree with the College of Artificial Intelligence, Nankai University, Tianjin, China.

Her research interest is deep learning.

Fangwei Hao is currently pursuing the Ph.D. degree with the College of Artificial Intelligence, Nankai University, Tianjin, China.

His main research focuses on image processing based on deep learning.

Ji Du is currently pursuing the Ph.D. degree with the College of Artificial Intelligence, Nankai University, Tianjin, China.

His main research focuses on computer vision.

Jing Xu (Member, IEEE) received the Ph.D. degree from Nankai University, Tianjin, China, in 2003.

She is a Professor at the College of Artificial Intelligence, Nankai University. She has published more than 100 papers in software engineering, software security, and big data analytics.

Dr. Xu won the Second Prize of Tianjin Science and Technology Progress Award twice in 2017 and 2018, respectively.

Ping Li (Member, IEEE) received the Ph.D. degree in computer science and engineering from The Chinese University of Hong Kong, Hong Kong, in 2013.

He is currently an Assistant Professor with the Department of Computing and an Assistant Professor with the School of Design, The Hong Kong Polytechnic University, Hong Kong. His current research interests include computer vision and creative media.