

Received 25 January 2024, accepted 19 March 2024, date of publication 22 March 2024, date of current version 28 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3380893

RESEARCH ARTICLE

Detecting Camouflaged Objects via Multi-Stage Coarse-to-Fine Refinement

YUYE WANG¹, TIANYOU CHEN², XIAOGUANG HU³, (Member, IEEE),
JIAQI SHI³, AND ZICHONG JIA³

¹College of Physics and Information Engineering, Minnan Normal University, Zhangzhou, Fujian 363000, China

²Faculty of Artificial Intelligence in Education, Central China Normal University, Wuhan, Hubei 430079, China

³State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing 100191, China

Corresponding author: Tianyou Chen (chentianyou@buaa.edu.cn)


This work was supported by the National Natural Science Foundation of China under Grant 51807003.

ABSTRACT Camouflaged objects are typically assimilated into their surroundings. Consequently, in contrast to generic object detection/segmentation, camouflaged object detection proves to be considerably more intricate due to the indistinct boundaries and heightened intrinsic similarities between foreground targets and the surrounding environment. Despite the proposition of numerous algorithms that have demonstrated commendable performance across various scenarios, these approaches may still grapple with blurred boundaries, leading to the inadvertent omission of camouflaged targets in challenging scenes. In this paper, we introduce a multi-stage framework tailored for segmenting camouflaged objects through a process of coarse-to-fine refinement. Specifically, our network encompasses three distinct decoders, each fulfilling a unique role in the model. In the initial decoder, we introduce the Bi-directional Locating Module to excavate foreground and background cues, enhancing target localization. The second decoder focuses on leveraging boundary information to augment overall performance, incorporating the Multi-level Feature Fusion Module to generate prediction maps with finer boundaries. Subsequently, the third decoder introduces the Mask-guided Fusion Module, designed to process high-resolution features under the guidance of the second decoder's results. This approach enables the preservation of structural details and the generation of fine-grained prediction maps. Through the integration of the three decoders, our model effectively identifies and segments camouflaged targets. Extensive experiments are conducted on three commonly used benchmark datasets. The results of these experiments demonstrate that, even without the application of pre-processing or post-processing techniques, our model outperforms 14 state-of-the-art algorithms.

INDEX TERMS Camouflaged object detection, coarse-to-fine refinement, convolutional neural network, multi-stage detection.

I. INTRODUCTION

In nature, numerous wild animals strive to seamlessly blend into their surroundings, adapting to the environment [1], [2], [3], [4]. Camouflage, as an effective technique for deceiving the observer's visual perceptual system, is widely adopted by prey to minimize the risk of detection by predators [5]. Targeting the identification of objects with a similar appearance to the background, Camouflaged Object Detection (COD) has garnered significant attention [6], [7], [8]. Serving as

The associate editor coordinating the review of this manuscript and approving it for publication was Hossein Rahmani .

a fundamental pre-processing approach, COD has not only captured growing research interest but has also catalyzed advancements in various computer vision tasks, such as polyp segmentation [9], lung infection segmentation [10], defect detection [11], recreational art [12], and transparent object detection [13].

Similar to other tasks (e.g., salient object detection [14], [15]), early COD methods [16], [17], [18], [19] primarily rely on hard-crafted features to generate prediction maps. However, as a highly challenging problem, COD necessitates a substantial amount of visual perception knowledge [8] to eliminate ambiguities arising from the intrinsic

similarities between foreground targets and their surrounding environment. Since these conventional methods can only extract low-level detailed cues (e.g, edge [19], and motion pattern [20]), when confronted with complex scenarios, they tend to produce inaccurate results or even overlook targets due to the absence of high-level contextual information.

Recently, with the rapid development of Convolutional Neural Networks (CNNs), researchers have endeavored to construct CNN-based models to address the COD problem. Diverging from early traditional hard-crafted methods, CNN-based approaches can simultaneously exploit low-level detailed cues and high-level semantic information. Consequently, CNN-based methods outperform their traditional counterparts by a significant margin and have become the mainstream solution.

Thanks to the development of large-scale benchmark datasets [1], [21], numerous CNN-based models [22], [23], [24], [25], [26] have been proposed. Despite the remarkable performance of these methods, there is still room for improvement. Firstly, the majority of existing methods primarily concentrate on the main body of the targets while neglecting the boundaries. As emphasized in [27], pixels situated at boundaries pose greater difficulty and are consequently more pivotal than others. Consequently, these methods frequently encounter challenges related to fuzzy boundaries. Secondly, most approaches [1], [2], [3], [6], [7], [8] adhere to the encoder-decoder framework, in which a single decoder is employed to consolidate multi-level features extracted from the encoder. Typically, during the decoder stage, high-level features are transmitted to shallower levels to pinpoint the targets and suppress background noise. However, as indicated in [28], high-level features may overlook certain object parts and details. Thus, the straightforward encoder-decoder framework can lead to a decline in performance.

To address the aforementioned problem, we develop a novel CNN-based framework, namely Multi-stage Coarse-to-fine Refinement Network (MCRNet). MCRNet can yield accurate results and demonstrates outstanding performance in various complex scenarios.

First, we introduce a Bi-directional Locating Module (BLM), comprising two independent blocks designed to exploit foreground and background information separately. It is noteworthy that the intersection area of the foreground and background delineates the contour of the camouflaged target. Consequently, by employing the BLM, we not only bilaterally explore foreground and background information to enhance target localization but also generate a boundary prediction map to facilitate subsequent optimization. Second, a Multi-level Feature Fusion Module (MFFM) is devised to facilitate the aggregation of multi-level features. In the process of feature aggregation, the complementary relationship between multi-level features is exploited to enhance overall feature quality. Additionally, the boundary prediction map is incorporated to augment overall performance. Moreover, as highlighted in [29], the effective receptive field size of a CNN is generally smaller than the theoretical value.

To mitigate this issue, we utilize multiple convolutional layers to significantly expand the receptive fields, thereby also aiding in the extraction of multi-scale information to enhance the model's resilience to scale variation [30].

By employing the BLMs and MFFMs, we construct two decoders. The first decoder, based on BLMs, is designed to pinpoint the target, while the second one, built on MFFMs, generates prediction maps with sharper boundaries. Previous researches [27], [31], [32], [33] have confirmed that utilizing multiple decoders for iterative refinement enhances performance. Therefore, drawing inspiration from [32], we adopt a bifurcated backbone network as the encoder. Features extracted from the two branches of the encoder are independently fed to the two decoders. It is crucial to note that the spatial resolution of features obtained from the two branches is relatively small. Consequently, structural information may be compromised during the subsampling process, such as convolution and pooling. To address this issue, we introduce a Mask-guided Fusion Module (MFM). The MFM takes the low-level high-resolution feature map and the results of the second decoder as inputs. By leveraging the prediction maps, we effectively suppress background noise in low-level features while preserving structural details. Through the incorporation of these key modules, the proposed MCRNet demonstrates the capability to produce high-quality results across a range of challenging scenarios.

To showcase the exceptional performance of MCRNet and substantiate the efficacy of the proposed modules, we conduct experiments on three widely adopted benchmark datasets. The experimental outcomes confirm the superiority of MCRNet. In summary, this paper makes four major contributions:

- We introduce the Bi-directional Locating Module, which exploits foreground and background information bilaterally to locate the camouflaged targets and deduce the initial position of their boundaries more effectively.
- We design the Multi-level Feature Fusion Module to extract multi-scale information, rendering the model resilient to scale variation. Moreover, the effective receptive field size is expanded, enabling the capture of richer contextual cues that contribute to the enhancement of COD performance.
- We develop the Mask-guided Fusion Module, which concurrently leverages semantic information from the generated estimated camouflaged regions and complementary structural details from the low-level encoder feature. By fully capitalizing on this complementarity, the proposed module suppresses background noises, yielding more accurate results.
- We conduct extensive experiments on 3 widely used COD datasets. The experimental results validate that MCRNet surpasses 14 state-of-the-art CNN-based algorithms across six universally agreed evaluation metrics, which validates the effectiveness of our proposed method.

II. RELATED WORKS

A. CAMOUFLAGED OBJECT DETECTION

COD aims to identify the camouflaged targets within their high-similarity surrounding environment. Similar to other computer vision tasks, COD methods can be categorized into two classes: traditional and deep-learning-based. Concretely, early methods are designed based on hard-crafted features (e.g, 3D convexity [34], motion boundary [20], and intensity features [35]). However, as highlighted in [36], these traditional algorithms are less robust and are prone to generate erroneous results in complex scenarios. Recently, owing to the availability of large-scale COD dataset [1], [21], an increasing number of deep-learning-based COD models have emerged.

Le et al. (ANet) [21] propose an anabranch model. The classification branch of the model is used to judge whether the input image contains camouflaged objects. The main branch is employed to segment the camouflaged targets. Fan et al. (SINet) [1] collect the currently largest COD dataset and develop a simple but effective framework for COD. Sun et al. (C²FNet) [22] employ a dual-branch global context module to exploit affluent contextual cues. Furthermore, an attention-induced cross-level fusion module is designed to integrate features. Mei et al. [8] introduce a bio-inspired framework, which can mimic the process of predation in the wild. Lv et al. (LSR) [37] argue that explicitly modeling the conspicuousness of camouflaged targets against the background is helpful in boosting the performance. A ranking-based model is then proposed to simultaneously localize, segment, and rank camouflaged targets. Li et al. (JCOD) [26] develop a paradigm of exploiting the contradictory information to enhance the detection ability of both COD and salient object detection (SOD). Zhuge et al. (CubeNet) [38] introduce χ connection to the widely used encoder-decoder framework. Wang et al. (D²CNet) [39] design a novel approach for COD. The proposed model simulates the observation process of human visual mechanisms and employs self-refine and cross-refine units to compute more accurate camouflaged maps. Liu et al. (MSCAFNet) [40] focus on learning multi-scale context-aware features. An enhanced receptive field module and a cross-scale feature fusion module are introduced to fully refine multi-level features and achieve sufficient interaction of multi-scale cues.

Different from these methods, some works try to introduce auxiliary cues to further improve the performance. Zhu et al. (TINet) [41] introduce a texture label to boost the COD performance. Besides, an interactive guidance framework is developed to capture the fuzzy boundaries and the texture differences. Ren et al. (TANet) [25] develop a texture-aware refinement module to emphasize the texture differences between the camouflaged objects and the surroundings. Besides, a boundary-consistency loss is proposed to explore the object detail structures. Chen et al. (BgNet) [42] design a Locating module to excavate local detailed cues and global contextual information.

Thus, when contextual information is deficient, the extracted local cues can provide informative information to identify the targets. Sun et al. (BGNet) [3] propose a boundary-guided model, which is competent to learn valuable and extra object-related boundary semantics to guide the detection process. Zhai et al. (DTCNet) [43] employ the spatial organization of textons in the foreground and background area as important cues for COD. He et al. (ELDNet) [44] develop a novel method to generate progressively polished boundary likelihood maps, which are then employed to guide the feature fusion of camouflaged objects. Chen et al. (BCNet) [2] use pseudo-3D convolution operations to investigate the complementary relationship between the camouflaged regions and the corresponding boundaries.

B. MULTI-STAGE DETECTION

The effectiveness of using multiple decoders to conduct coarse-to-fine refinement has been validated by many methods. Generally, early algorithms tend to employ several decoders with the same architecture to iteratively polish features extracted from the encoder. Wei et al. (F³Net) [27] design the cross feature module to adaptively select complementary cues from input multi-level features, which is beneficial to avoid introducing redundant information that might bring negative impacts on the final performance. Furthermore, the cascaded feedback decoder (CFD) is developed. CFD consists of multiple sub-decoders with the same structure. Thus, the input features can be polished iteratively to obtain better performance. Wu et al. (CPD) [32] design the bifurcated encoder and build two decoders to aggregate features. A cascaded optimization mechanism is also developed to further improve the performance. Zhai et al. (BBSNet) [45] devise a cascaded refinement network, where multi-level features are split into two groups. Afterward, two decoders are employed to respectively process the two groups of features. Chen et al. (AFNet) [14] develop the cascaded feature interweaved decoder to investigate the complementary cues between input features and refine them iteratively to yield fine-grained prediction maps.

Recently, some works have found that using multiple decoders with different architectures may lead to better performance. Chen et al. (BgNet) [42] adopt the bifurcated encoder and design two decoders. In the first decoder, a locating module is utilized to generate a coarse segmentation result and a boundary prediction map, which are then fed to the second decoder. By leveraging the results of the first decoder, the second decoder can generate prediction maps with finer details. Fan et al. (SINetV2) [4] develop a neighbor connection decoder to locate the potential targets. The second decoder is composed of multiple group-reversal attention modules, which can effectively reproduce the identification stages of wild animal predation. However, in these methods, only high-level features are utilized to generate prediction maps. It is worth noting that high-level features lack structural details due to the multiple subsampling operations in the encoder. Thus, the performance might be degraded.

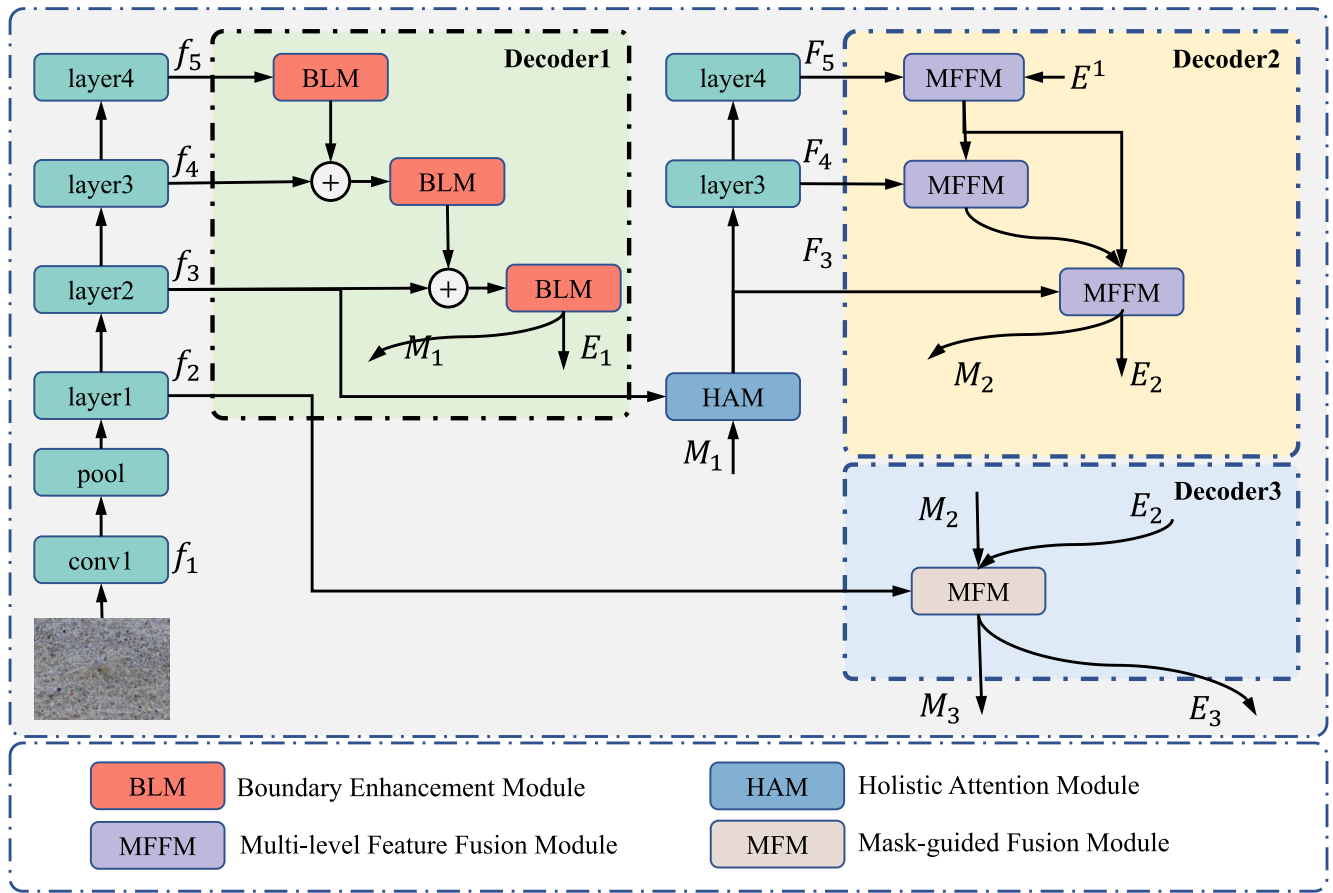


FIGURE 1. The overall pipeline of our proposed MCRNet.

III. METHODOLOGY

A. OVERVIEW OF THE PROPOSED MCRNET

The overall pipeline of the proposed MCRNet is illustrated in Figure 1. As depicted in the figure, MCRNet comprises a bifurcated backbone encoder and three decoders. When presented with an input image having a spatial resolution of $H \times W$, the encoder initially extracts features at five levels, denoted as $\{f_i; i = 1, 2, 3, 4, 5\}$, respectively. Subsequently, f_3, f_4, f_5 are input into the first decoder to produce a coarse result and a boundary prediction map. In this initial decoding stage, the primary emphasis is on pinpointing the target. As emphasized in [32], low-level features make a lesser contribution to the final performance and significantly amplify computational and memory overhead. Consequently, only the three high-level features are employed in the first decoder. The outcomes of the first decoder are computed as follows:

$$f'_5, m_5^1, e_5^1 = BLM(f_5), \tag{1}$$

$$f'_4, m_4^1, e_4^1 = BLM(f_4 + f'_5), \tag{2}$$

$$f'_3, M_1, E_1 = BLM(f_5 + f'_4), \tag{3}$$

where BLM denotes the BLM, $\{M_1, m_4, m_5\}$ are segmentation results, $\{E_1, e_4, e_5\}$ are boundary prediction maps. Note that

we use bilinear interpolation operation for upsampling and all upsampling processes are omitted for conciseness.

After acquiring the preliminary results, we can utilize the prediction maps to eliminate background noises in encoder features. Nevertheless, the outcomes from the initial decoder typically exhibit coarseness in boundary delineation. Directly multiplying an encoder feature with the segmentation result might inadvertently remove edge information related to camouflaged targets, resulting in a decline in performance. Therefore, we incorporate the Holistic Attention module (HAM) [32]. Specifically, HAM takes the feature map f_3 and the prediction map of the first decoder M_1 as inputs. The output of HAM can be obtained by computing:

$$F_3 = f_3 \times MAX(f_{minmax}(Conv_g(M_1, k)), M_1), \tag{4}$$

where $Conv_g$ denotes a convolution operation with a Gaussian kernel k and 0 bias, f_{minmax} is a normalization function, MAX is used to compare $Conv_g(M_1, k)$ and M_1 to get the larger result. By employing HAM, we can highlight the entire camouflaged regions.

The enhanced feature F_3 is then propagated to the second branch of the encoder. Afterward, we can obtain three high-level features $\{F_i; i = 3, 4, 5\}$, which are then fed to the second decoder to calculate the finer results. The process

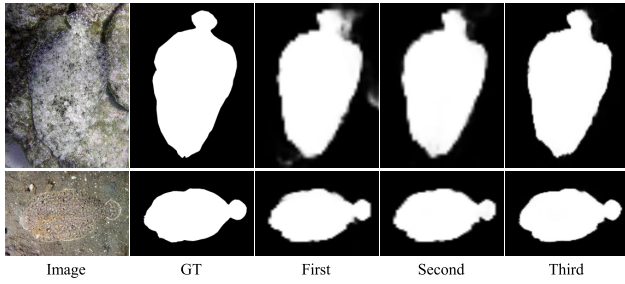


FIGURE 2. The results of different decoders.

can be formulated as:

$$F'_5, m_5^2, e_5^2 = MFFM(F_5, E_1), \quad (5)$$

$$F'_4, m_4^2, e_4^2 = MFFM(F_4, F'_5, e_5^2), \quad (6)$$

$$F'_3, M_2, E_2 = MFFM(F_3, F'_4, F'_5, e_4^2), \quad (7)$$

where *MFFM* denotes the MFFM, M_2 and E_2 are the output segmentation result and the boundary prediction map, respectively.

As depicted in Figure 2, the output of the second decoder may exhibit blurring at its boundaries. We posit that this phenomenon can be partly ascribed to the subsampling operations employed in the encoder. Given the relatively small size of the input features for the two decoders, there exists a potential loss of structural information. While it is possible to upscale the generated prediction map to attain the full size, the retrieval of the lost information remains incomplete. Consequently, we introduce the MFM, which takes the high-resolution feature f_2 and incorporates the outputs of the second decoder as input. This process is expressed as

$$M_3, E_3 = MFM(f_2, M_2, E_2), \quad (8)$$

where M_3 and E_3 are the final result.

The proposed MCRNet can be trained in an end-to-end manner. Without bells and whistles, MCRNet consistently produces accurate results across diverse challenging scenarios. Besides, the MCRNet can achieve a real-time inference speed of 31 FPS on a workstation with a single NVIDIA Titan XP GPU card.

B. BI-DIRECTIONAL LOCATING MODULE

The structure of the proposed BLM is illustrated in Figure 3. For an input feature f_{in} , we first use a 1×1 convolutional layer for channel compression, which is effective in reducing the computational and memory overhead. Subsequently, the compressed feature f is fed to two independent blocks, namely the foreground block and the background block. It is noteworthy that the architectures of these two blocks are identical. Consequently, the details of the background block are omitted in Figure 3 for simplicity.

The foreground block is composed of four branches, each containing a 3×3 convolutional layer. Following the application of the 3×3 convolutional layers, the outcomes $\{f_c^i; i = 1, 2, 3, 4\}$ are acquired. In the initial branch, a

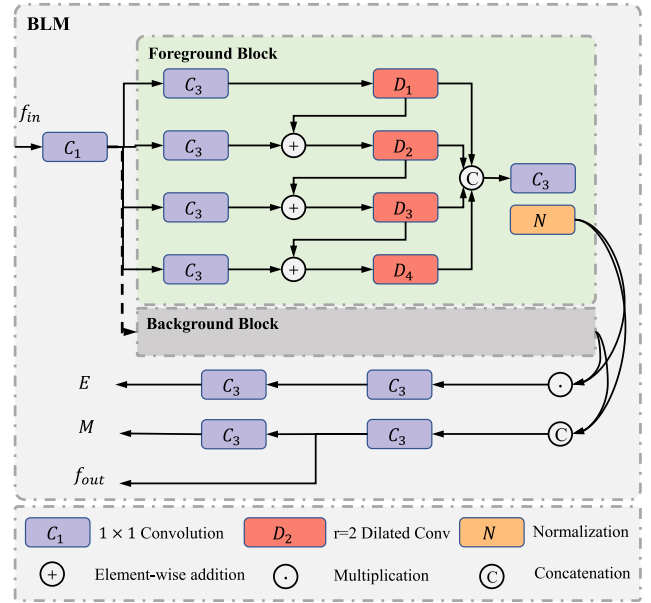


FIGURE 3. The structure of the proposed BLM.

3×3 dilated convolutional layer with a dilation rate of 1 is employed. Subsequently, the outcome of the first branch is fused with f_c^2 through element-wise addition. For further refinement, a 3×3 dilated convolutional layer with a dilation rate of 2 is implemented in the second branch. The result of the second branch undergoes the same process in the third branch. This iterative process facilitates the continuous refinement of the input feature, significantly expanding the receptive fields. Upon obtaining the outcomes from the four branches, a channel-wise concatenation is performed to integrate them. The concatenated feature undergoes processing through a 3×3 convolutional layer. Subsequently, a normalization function is applied to constrain the feature values within the range of $[0, 1]$. The entire process can be expressed as follows:

$$f_d^i = \begin{cases} D_i(f_c^i), & i = 1, \\ D_i(f_c^i + f_d^{i-1}), & i = 2, 3, 4, \end{cases} \quad (9)$$

$$f_n = f_{minmax}(C_3(cat(f_d^1, f_d^2, f_d^3, f_d^4))), \quad (10)$$

where D_i denotes a 3×3 convolutional layer with a dilation rate of i , f_d^i represent the output of the i branch, *cat* is the concatenation operation, C_3 denotes a 3×3 convolutional layer, f_n is the output of the normalization function.

Similarly, we can derive the output of the background block, denoted as b_n . It is important to observe that the boundaries of the camouflaged targets coincide with the intersection area of the foreground and background regions. Consequently, the boundary prediction map can be obtained by computing:

$$E = C_3(C_3(f_n \times b_n)). \quad (11)$$

As pointed out in [28], exploring potential targets in background regions is helpful in improving the performance.

Thus, we concatenate the foreground and background features along the channel dimension. The fused feature is then fed to a 3×3 convolutional layer for channel compression and refinement. Then, a 3×3 convolutional layer is utilized to generate the segmentation result. The whole process can be formulated as:

$$f_{out} = C_3(cat(f_n, b_n)), \quad (12)$$

$$M = C_3(f_{out}), \quad (13)$$

where f_{out} is the output feature, M is the segmentation result.

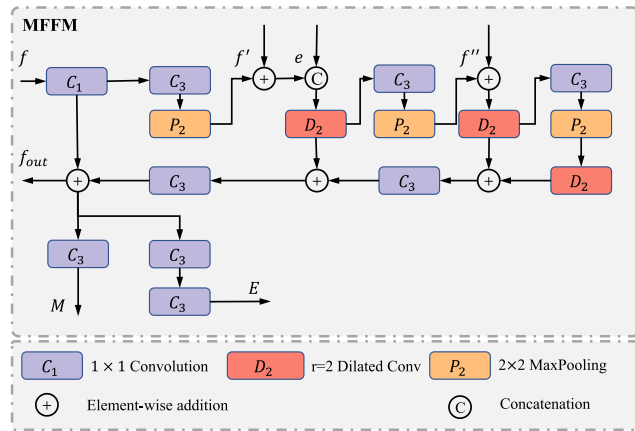


FIGURE 4. The structure of the proposed MFFM.

C. MULTI-LEVEL FEATURE FUSION MODULE

The structure of the proposed MFFM is shown in Figure 4. In the figure, f denotes the side-output encoder feature, while f' and e respectively represent the output feature and the boundary prediction map of the preceding MFFM. Furthermore, f'' corresponds to the output feature of the initial MFFM.

As illustrated in the figure, the MFFM encompasses a left-to-right process and a right-to-left process. In the left-to-right process, multi-level features are aggregated, and the receptive fields are expanded. More specifically, the left-to-right process can be delineated into three stages. In each stage, we initially apply a 3×3 convolutional layer followed by a 2×2 max-pooling layer to exploit multi-scale information and augment the receptive fields. Subsequently, the resultant feature is integrated with other features or prediction maps. Following this, a 3×3 dilated convolutional layer with a dilation rate of 2 is employed for further refinement. The entire process can be formulated as follows:

$$f^0 = C_1(f), \quad (14)$$

$$f^i = \begin{cases} P_2(C_3(f^{i-1})), & i = 1, \\ P_2(C_3(f_d^{i-1})), & i = 2, 3, \end{cases} \quad (15)$$

$$f_d^i = \begin{cases} D_2(cat(f^i + f', e)), & i = 1, \\ D_2(f^i + f''), & i = 2, \\ D_2(f^i), & i = 3. \end{cases} \quad (16)$$

While the left-to-right process can integrate multi-level features, structural information is lost due to the multiple subsampling operations. In contrast, in the right-to-left process, we progressively upsample the features and integrate them with higher resolution features to recover structural details. The process can be depicted as:

$$f_a^i = \begin{cases} f^0 + C_3(f_a^2) & i = 1, \\ f_d^i + C_3(f_a^{i+1}) & i = 2, \\ f_d^i + f_d^{i-1} & i = 3, \end{cases} \quad (17)$$

where $f_{out} = f_a^1$ is the output feature.

As indicated in [30], [46], and [47], low-level features capture rich spatial information and effectively highlight boundaries. However, they also contain abundant background noise. In contrast, high-level features encode semantic knowledge, thus facilitating more accurate target localization. Through the integration of multi-level features, we can effectively eliminate background noise while preserving intricate spatial details. Additionally, multi-scale information is leveraged to enhance the model's robustness to scale variation. It is worth noting that the majority of convolution operations in MFFM are performed on subsampled features, resulting in a minimal computational overhead for the module.

After obtaining f_{out} , we can generate the segmentation result and the corresponding boundary prediction map by computing:

$$M = C_3(f_{out}), E = C_3(C_3(f_{out})). \quad (18)$$

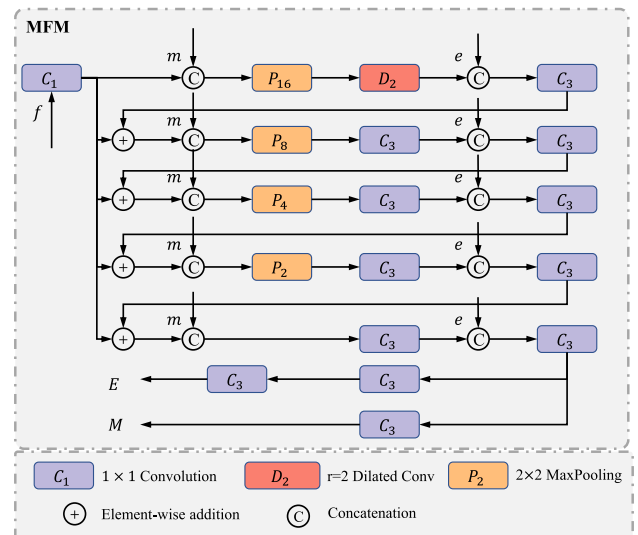


FIGURE 5. The structure of the proposed MFM.

D. MASK-GUIDED FUSION MODULE

The structure of our proposed MFM is illustrated in Figure 5. The MFM model takes the results from the second decoder and the encoder feature as inputs. In the figure, the

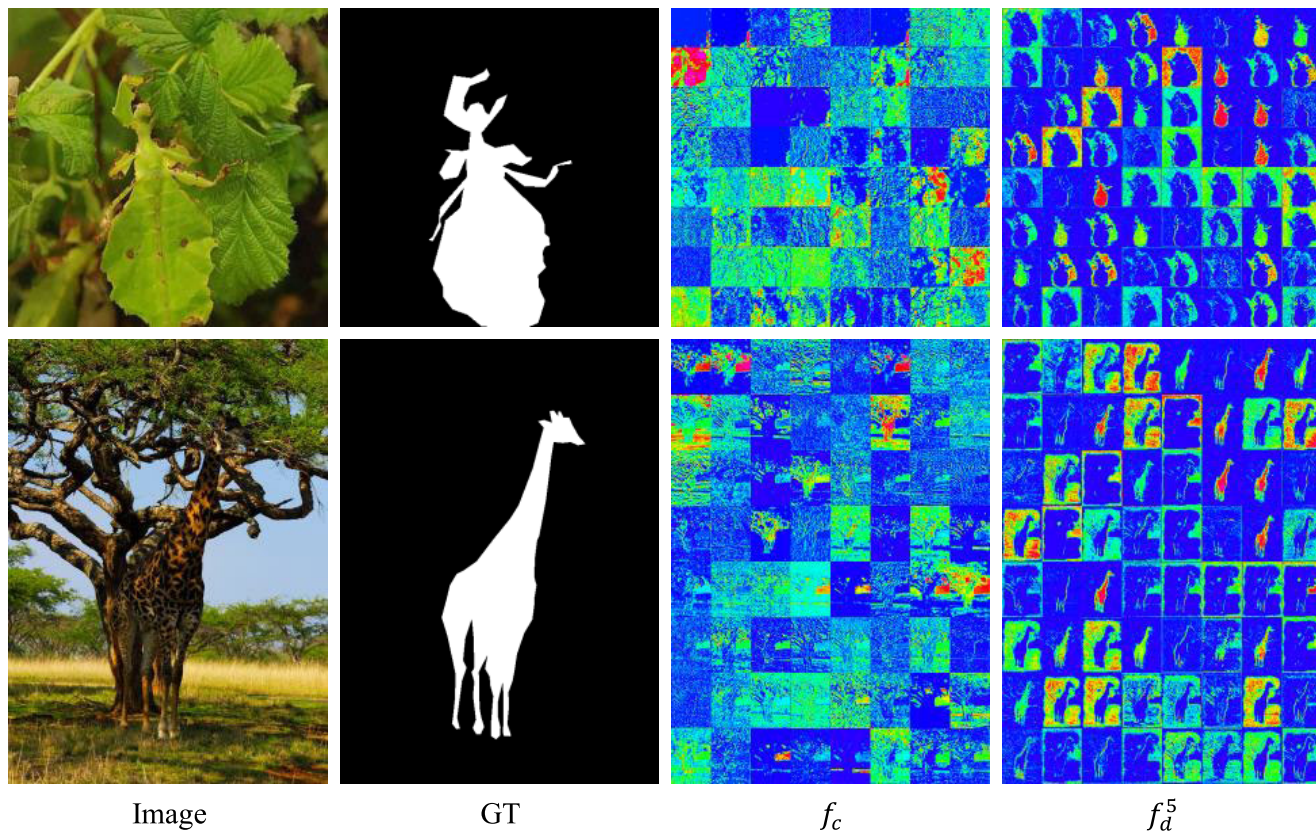


FIGURE 6. Per-channel visualization results. Introducing MFM allows the MCRNet to eliminate background noises and precisely locate the target.

encoder feature is denoted by f , while m and e represent the segmentation result and the boundary prediction map, respectively.

We initiate the process by applying a 1×1 convolutional layer to reduce the channel number of f . As depicted in the figure, MFM comprises five branches. It is crucial to note that, being a low-level feature, f generally contains significant background noise. Therefore, in the first branch, we initially combine the feature with m through channel-wise concatenation. Subsequently, a 16×16 maxpooling layer is employed to decrease the spatial resolution. This step is essential as it allows for the utilization of a dilated convolutional layer to more effectively capture global contextual information and eliminate background noise in the feature map. Following this, the resultant feature undergoes upsampling and is concatenated with e , which enhances the generation of results with sharper boundaries. It is important to mention that the upsampling process is not depicted in the figure for the sake of conciseness. Subsequently, a 3×3 convolutional layer is employed for refinement. The entire process can be formulated as follows:

$$f_c = C_1(f), \tag{19}$$

$$f_c^1 = D_2(P_{16}(cat(f_c, m))), \tag{20}$$

$$f_b^1 = C_3(cat(f_c^1, e)), \tag{21}$$

where f_b^1 is the result of the first branch.

Although the first branch can help alleviate the negative impacts of background interference, the 16×16 max-pooling operation may result in a loss of details. Hence, we combine f_b^1 with f_c to recover the lost details. In contrast to the first branch, we employ P_8 in the second branch to generate features with finer details. Additionally, we utilize a standard convolution operation for refinement. By progressively reducing the kernel sizes of the pooling layers, we can effectively eliminate background noises while preserving rich spatial details. The results of the five branches can be expressed as:

$$f_c^i = \begin{cases} D_2(P_{16}(cat(f_c, m))), & i = 1, \\ D_2(P_{2^{5-i}}(cat(f_c + f_d^{i-1}, m))), & i = 2, 3, 4, 5, \end{cases} \tag{22}$$

$$f_d^i = \begin{cases} C_3(cat(f_c^i, e)), & i = 1, 2, 3, 4, 5, \end{cases} \tag{23}$$

where P_1 is an identical mapping, f_d^i is the result of the i -th branch.

Then, the final segmentation result M and the boundary prediction map E can be calculated as:

$$M = C_3(f_d^5), E = C_3(C_3(f_d^5)). \tag{24}$$

To demonstrate the effectiveness of the proposed MFM, we visualize some feature maps. As shown in Figure 6, by comparing f_c with f_d^5 , it can be clearly found that

TABLE 1. Comparison of the proposed MCRNet with 14 high-performance models on 3 COD benchmark datasets in terms of 4 evaluation metrics. The best results are shown in **‘‘**. ‘‘ indicates that both the segmentation results and the evaluation scores are not available.

Method	Backbone	CAMO				COD10K				NC4K			
		S_m	F_β^w	M	E_ϕ	S_m	F_β^w	M	E_ϕ	S_m	F_β^w	M	E_ϕ
SINet [1]	ResNet-50	.751	.606	.100	.771	.771	.551	.051	.806	.808	.723	.058	.871
ERRNet [48]	ResNet-50	.747	.667	.087	.849	.739	.589	.048	.868	.783	.704	.060	.887
CubeNet [38]	ResNet-50	.788	.682	.085	.838	.795	.644	.041	.864	-	-	-	-
TANet [25]	ResNet-50	.778	.659	.089	.813	.794	.613	.043	.838	-	-	-	-
TINet [41]	ResNet-50	.781	.678	.087	.847	.793	.635	.043	.848	-	-	-	-
DTCNet [43]	ResNet-50	.778	.667	.084	.804	.790	.616	.041	.821	-	-	-	-
PFNet [49]	ResNet-50	.782	.695	.085	.842	.800	.660	.040	.877	.829	.745	.053	.888
MCRNet	ResNet-50	.810	.727	.075	.866	.806	.658	.040	.879	.840	.752	.051	.894
PraNet [9]	Res2Net-50	.769	.663	.094	.825	.789	.629	.045	.861	.822	.724	.059	.876
BSANet [23]	Res2Net-50	.794	.717	.079	.851	.818	.699	.034	.891	.841	.771	.048	.897
C ² FNet [22]	Res2Net-50	.796	.719	.080	.854	.813	.686	.036	.890	.838	.762	.049	.897
FAPNet [7]	Res2Net-50	.817	.734	.076	.865	.822	.694	.036	.888	.851	.775	.047	.899
SINetV2 [4]	Res2Net-50	.820	.743	.070	.882	.815	.680	.037	.887	.847	.770	.048	.903
TSFNet [50]	Res2Net-50	.822	.759	.068	.884	.825	.704	.034	.897c	.853	.787	.044	.909
BGNet [3]	Res2Net-50	.813	.749	.073	.870	.831	.722	.033	.901	.851	.788	.044	.907
MCRNet+	Res2Net-50	.831	.761	.064	.891	.834	.706	.032	.902	.860	.790	.044	.910

introducing MFM allows the MCRNet to eliminate background noises and precisely locate the target.

E. LOSS FUNCTION

As done in many previous methods [4], [42], we adopt the hybrid loss function [27], which is defined as:

$$L = L_{IoU}^w + L_{BCE}^w, \quad (25)$$

where L_{BCE}^w is the weighted binary cross entropy (wBCE) loss and L_{IoU}^w is the weighted intersection-over-union (wIoU) loss. The standard BCE loss allocates equal weights to all pixels. Nevertheless, pixels situated at boundaries or within elongated regions pose greater difficulty and warrant increased significance. In response to this challenge, both wBCE and wIoU losses augment the weights assigned to hard pixels. The efficacy of these loss functions has been substantiated in numerous prior studies [4], [8], [42]. In this context, we incorporate the deep supervision strategy for the outcomes produced by the three decoders. The total loss for the proposed MCRNet is formulated as follows:

$$L_t = \sum_{i=1}^3 L(M_i, M_G) + \sum_{i=1}^3 L(E_i, E_G), \quad (26)$$

where M_i and E_i are respectively segmentation result and boundary prediction map of the i decoder, M_G and E_G are the groundtruth.

IV. EXPERIMENTS

A. DATASETS AND EVALUATION METRICS

We conduct experiments to evaluate MCRNet on 3 widely used COD datasets, i.e., CAMO [21], COD10K [1], and NC4K [37]. CAMO contains 1.25K precisely annotated images, which are divided into a training subset (1K images) and an evaluation subset (0.25k images). COD10K is the currently largest dataset for COD, which is composed of 5,066 images featuring various camouflaged objects in challenging scenarios. It is divided into a training subset of

3,040 samples and a testing subset of 2,026 images. NC4K has 4,121 high-quality samples gathered from the Internet. As a dataset with the most images for evaluation, it can better reveal the generalization ability of COD models.

To better evaluate the performance, we adopt six widely used metrics, including 1) S-measure (S_m) [51], 2) weighted F-measure (F_β^w) [52], 3) mean absolute error (M), 4) mean E-measure (E_ϕ) [53], 5) Precision-Recall curves, 6) F-measure curves. For M , a higher value indicates worse performance. For other evaluation metrics, higher is better.

(1) *Precision-Recall curves (PR-curves)*: By employing a threshold ranging from 0 to 255, the produced segmentation results can be converted into binary maps. Consequently, precision and recall scores for a prediction map can be calculated by comparing the corresponding binarized map with the groundtruth. Additionally, when applied to a benchmark dataset, precision and recall scores can be acquired by averaging the scores of all segmentation results. As a result, a series of average precision-recall pairs can be computed by adjusting the threshold. Subsequently, PR-curves can be plotted.

(2) *F-measure curves*: As a harmonic average of precision and recall, F-measure can be obtained by computing:

$$F_\beta = \frac{(1 + \beta^2) \times Precision \times Recall}{\beta^2 \times Precision + Recall}, \quad (27)$$

where $\beta^2 = 0.3$ to highlight precision as done in [4] and [8]. Based on the precision-recall pairs, we compute a series of F-measure values and plot the F-measure curves.

(3) *S-measure (S_m)* [51]: S_m is employed to judge the structural similarity of the segmentation result and the groundtruth. It can be obtained by computing the weighted sum of S_r (i.e., region-aware structural similarity) and S_o (i.e., object-aware structural similarity) as follows:

$$S_m = (1 - \alpha)S_r + \alpha S_o, \quad (28)$$

where $\alpha = 0.5$ as done in [4], [8], [22], and [42].

(4) *weighted F-measure (F_β^w)* [52]: As shown in many works [51], [53], the weighted F-measure [52] has been proven more reliable than the traditional F-measure when used for evaluation. Thus, we adopt F_β^w as a supplementary evaluation metric.

(5) *Mean absolute error (M)*: The mean absolute error is widely used in many image segmentation tasks. It reveals the per-pixel difference between a segmentation result and its groundtruth.

(6) *E-measure (E_ϕ)* [53]: By leveraging image-level statistics and local pixel-level matching, E-measure can be used to evaluate the similarity between a segmentation result and the corresponding groundtruth label. We report the mean E_ϕ as done in [1], [4], [8], and [42].

B. IMPLEMENTATION DETAILS

We implement MCRNet utilizing the PyTorch toolbox. As the most widely used backbone network in COD, ResNet-50 [54] is adopted to build the bifurcated encoder. Throughout the training phase, all images are resized to 384×384 and augmented using multiple strategies (e.g., horizontal flipping, random rotating, and border clipping) to mitigate the risk of overfitting [45]. We set the batch size to 16 and adopt Adam optimizer [55] with an initial learning rate of $4e-5$ for model optimization. The learning rate is divided by 10 every 40 epochs. MCRNet is trained for 120 epochs and the whole training process consume approximately 15 hours on a single NVIDIA Titan XP GPU. During testing, images are also resized to 384×384 . We only leverage the segmentation result of the last decoder for evaluation.

C. COMPARISONS TO THE STATE-OF-THE-ARTS

We compare our proposed MCRNet with 14 state-of-the-art methods including SINet [1], ERRNet [48], CubeNet [38], TANet [25], TINet [41], DTCNet [43], PFNet [8], PraNet [9], BSAntet [23], C²FNet [22], FAPNet [7], SINetV2 [4], TSFNet [50] and BGNet [3]. The evaluation results of SINet and C²FNet on NC4K dataset is not provided. Thus, we run the released codes with pretrained models for evaluation. Both the source codes and the segmentation results of some methods (e.g., TINet, DTCNet, CubeNet and TANet) on NC4K are not available, thus we do not present their results on NC4K. Since some competing methods are built on Res2Net-50 [56], we construct MCRNet+ using the same backbone to ensure a fair comparison.

The quantitative experimental results of MCRNet and all contenders are demonstrated in Table 1. As shown in the table, the proposed MCRNet outperforms all contenders in terms of all evaluation metrics on the 3 benchmark datasets. Concretely, performance gains over the best counterparts built on ResNet-50 backbone network (i.e., PFNet) are (0.6% ~ 2.8%, 0% ~ 3.2%, 0 ~ 0.010, 0.2% ~ 2.4%) for metrics (S_m , F_β^w , M , E_ϕ). Furthermore, when employing Res2Net-50 [56] as the backbone encoder, our MCRNet+ shows better performance, and surpasses the best contender

(i.e., BGNet) by (0.3% ~ 1.8%, 0 ~ 0.009, 0.1% ~ 2.1%) for the metrics (S_m , M , E_ϕ). The PR-curves and F-measure curves of the MCRNet and 8 state-of-the-art competing models are illustrated in Figure 7 and Figure 8 to provide a more comprehensive evaluation. As we can observe from the figures, MCRNet (red lines) demonstrate the best performance.

We present several representative segmentation results of the MCRNet and eight state-of-the-art models in Figure 9. Specifically, the first and second rows display the prediction maps of images with large camouflaged targets. As evident from the figure, MCRNet excels in capturing the entire targets, whereas other contenders tend to neglect certain parts of the targets. This observation substantiates that the proposed model effectively extracts rich contextual information. Consequently, MCRNet demonstrates superior performance when confronted with images featuring large objects.

In the third and fourth rows, the targets are considerably small in size. While competing methods mistakenly classify some background regions as camouflaged objects, our method produces more reliable results. The fifth and sixth rows depict images with multiple targets, while the seventh and eighth rows showcase results from images with intricate shapes. As evident from the figure, the proposed MCRNet not only captures the entire targets but also accurately segments thin structures (e.g., spider legs), which are overlooked by other algorithms. This demonstrates that the proposed model preserves detailed structure cues more effectively.

The ninth and tenth rows present two challenging cases where the internal texture of the camouflaged targets closely resembles the background. While state-of-the-art contenders are confused by the background, the proposed MCRNet precisely locates the targets and accurately segments the camouflaged regions. In summary, MCRNet adeptly identifies camouflaged regions and produces fine-grained segmentation results under various highly challenging scenarios.

D. ABLATION STUDY

We conduct ablation experiments on two datasets (i.e., CAMO and COD10K) to validate the effectiveness of the key modules.

1) PERFORMANCE OF DIFFERENT DECODERS

We compare the performance of the three decoders to demonstrate the effectiveness of our coarse-to-fine refinement strategy. The results are shown in Table 2. As can be seen

TABLE 2. Performance of different decoders.

	CAMO				COD10K			
	S_m	F_β^w	M	E_ϕ	S_m	F_β^w	M	E_ϕ
First	.805	.697	.081	.851	.789	.607	.045	.848
Second	.808	.705	.078	.856	.795	.618	.044	.854
Third	.810	.727	.075	.866	.806	.658	.040	.879

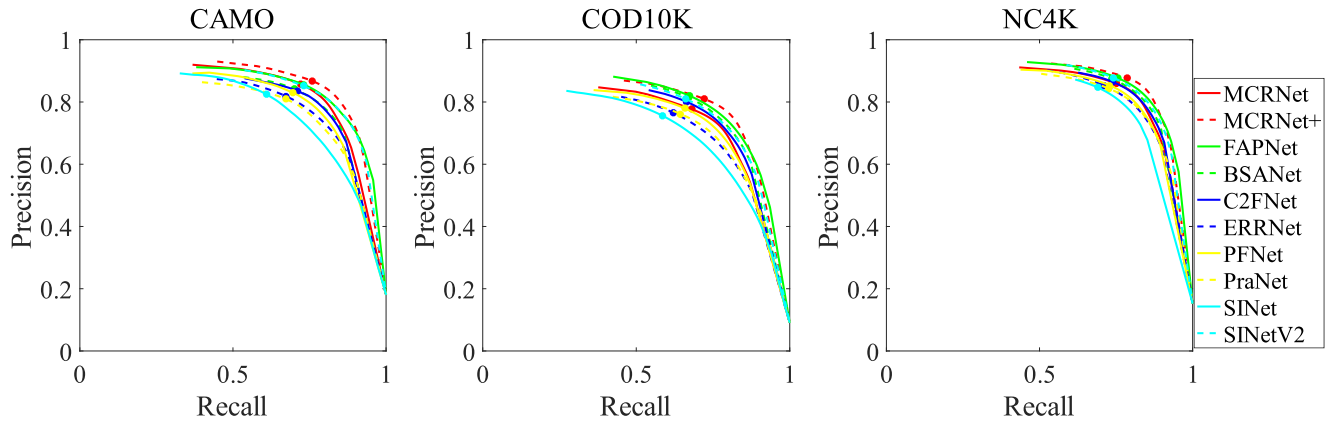


FIGURE 7. Precision-recall curves of the proposed MCRNet and 8 state-of-the-art methods.

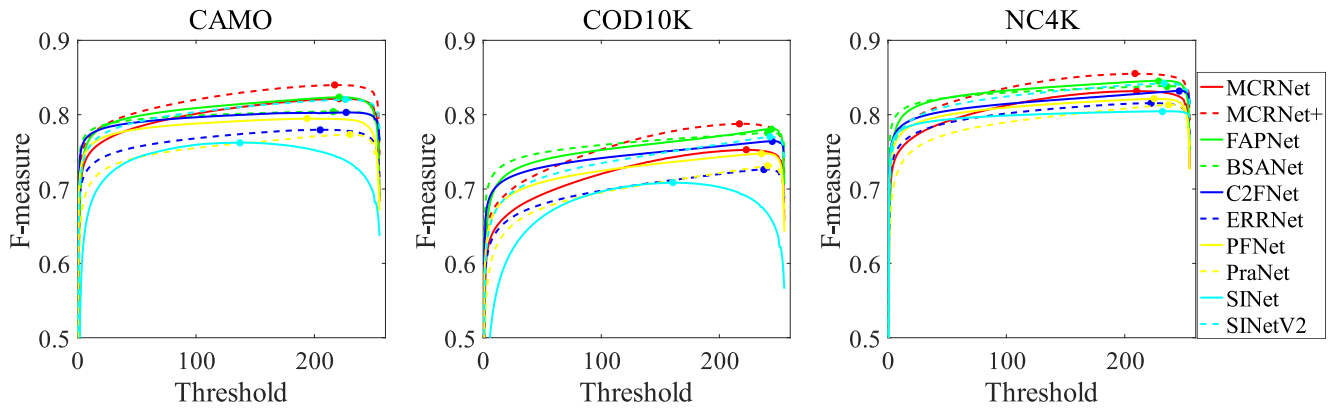


FIGURE 8. F-measure curves of the proposed MCRNet and 8 state-of-the-art methods.

TABLE 3. Ablation analysis for the proposed BLM.

	CAMO				COD10K			
	S_m	F_β^w	M	E_ϕ	S_m	F_β^w	M	E_ϕ
w/o Back	.806	.720	.078	.854	.792	.634	.046	.864
w/o MS	.808	.723	.076	.851	.802	.652	.042	.873
MCRNet	.810	.727	.075	.866	.806	.658	.040	.879

from the table, the last decoder shows the best performance. Besides, the second decoder also outperforms the first one by a significant margin. These experimental results verify that using multiple decoder for iterative refinement is beneficial to improve the performance.

2) EFFECTIVENESS OF THE BLM

To validate the effectiveness of the proposed BLM, we conduct ablation experiments. Specifically, we train two versions, namely “w/o Back” and “w/o MS”. In “w/o Back”, we remove the background block in BLM. In “w/o MS”, we remove the four branches of convolution operations in foreground and background blocks. The quantitative evaluation results are shown in Table 3.

As can be seen from the table, the performance degradation of “w/o Back” is (S_m : 0.4% ~ 1.4%, F_β^w : 0.7% ~ 2.4%, M : 0.003 ~ 0.006, E_ϕ : 1.2% ~ 1.5%), which demonstrates that bilaterally exploring informative cues can boost the performance. This is because that the foreground feature may not capture the entire target. Exploring the potential camouflaged objects in background regions can help mine neglected targets. Besides, comparing “w/o MS” with MCRNet suggests that excavating multi-scale information is effective in improving the performance.

3) EFFECTIVENESS OF THE MFFM

To verify the effectiveness of the MFFM, we train two variants, namely “MFFM-WM” and “MFFM-WE”. In “MFFM-WM”, we first directly aggregate multi-level feature via element-wise addition. Then, the convolution operations used for multi-scale information extraction are removed. In “MFFM-WE”, the MFFM does not take boundary prediction map as input. Besides, it does not generate boundary prediction map as well. The evaluation results are available at Table 4.

As can be seen from the table, MCRNet surpasses “MFFM-WM” by a large margin. The performance

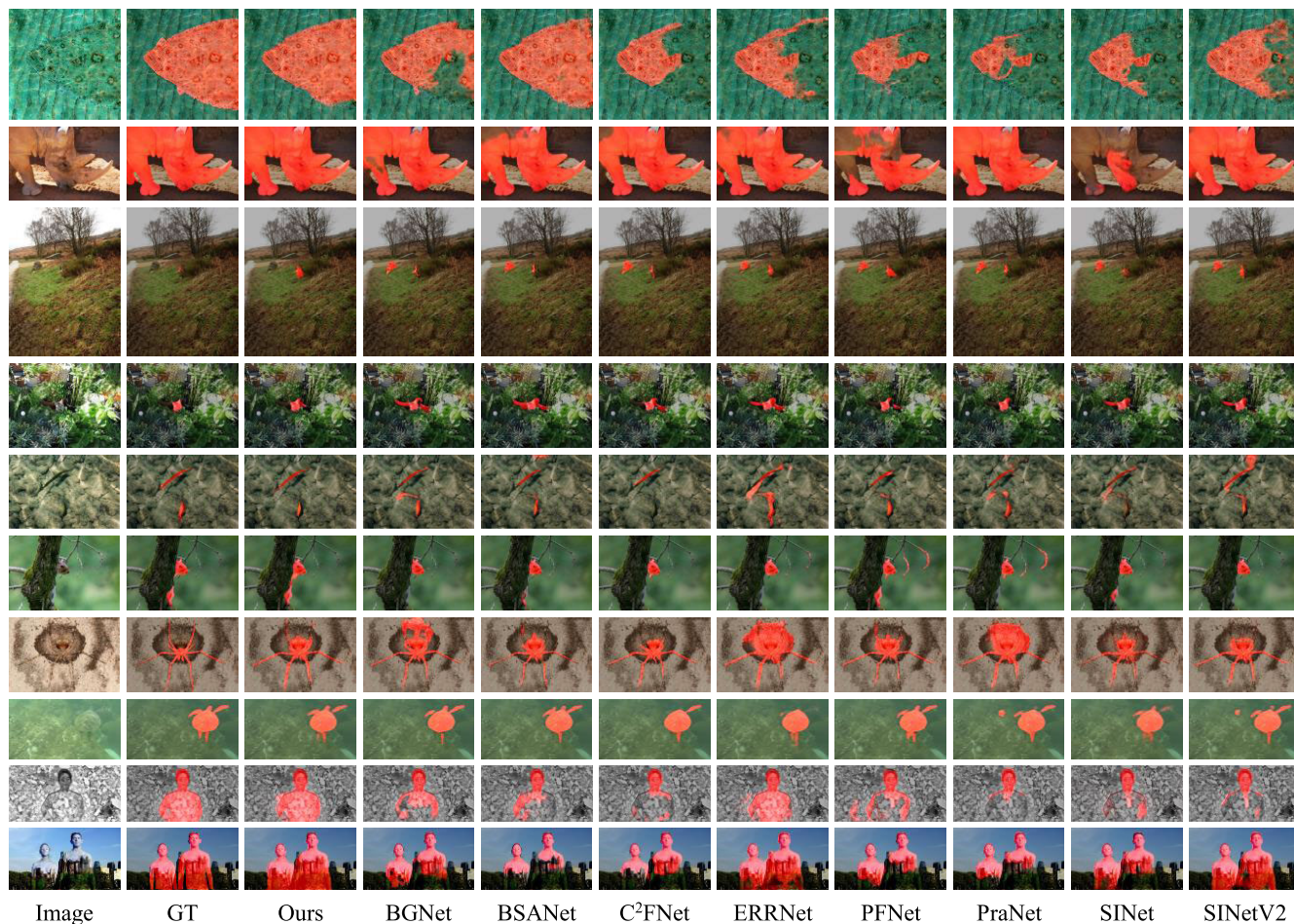


FIGURE 9. Qualitative comparisons of the proposed MCRNet with 8 state-of-the-art methods.

TABLE 4. Ablation analysis for the proposed MFFM.

	CAMO				COD10K			
	S_m	F_β^w	M	E_ϕ	S_m	F_β^w	M	E_ϕ
MFFM-WM	.795	.698	.081	.842	.796	.642	.043	.868
MFFM-WE	.806	.720	.078	.861	.799	.651	.041	.872
MCRNet	.810	.727	.075	.866	.806	.658	.040	.879

degradation of “MFFM-WM” is (S_m : 1.0% ~ 1.5%, F_β^w : 1.6% ~ 2.9%, M : 0.003 ~ 0.006, E_ϕ : 1.1% ~ 2.4%), which verify that exploiting multi-scale information is effective in boosting the performance. The experimental results are also consistent with previous observations in Section IV-D2. The performance of “MFFM-WE” is also degraded (i.e., S_m : 0.4% ~ 0.7%, F_β^w : 0.7%, M : 0.001 ~ 0.003, E_ϕ : 0.5% ~ 0.7%), which proves that incorporating boundary information is helpful in generating segmentation results with finer details.

4) EFFECTIVENESS OF THE MFM

We implement two variants of MCRNet to reveal to the effectiveness of MFM. Concretely, the first variant is denoted

TABLE 5. Ablation analysis for the proposed MFM.

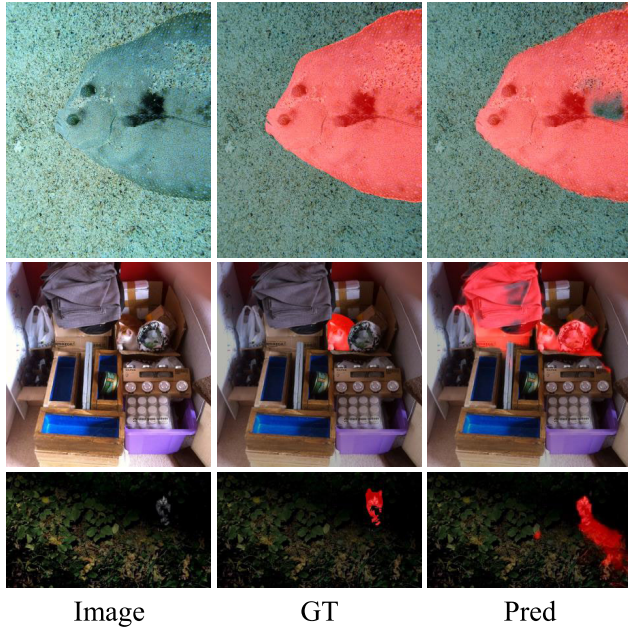
	CAMO				COD10K			
	S_m	F_β^w	M	E_ϕ	S_m	F_β^w	M	E_ϕ
MFM-WE	.793	.698	.081	.849	.799	.647	.041	.870
MFM-WM	.797	.703	.080	.858	.796	.639	.042	.867
MCRNet	.810	.727	.075	.866	.806	.658	.040	.879

as “MFM-WE”. This model does not take the boundary prediction map of the second decoder as input. The second decoder is denoted as “MFM-WM”. This variant removes the four branches of the MFM and only employs the last branch to integrate high-resolution feature with the prediction maps. The experimental results are presented in Table 5.

As we can observe, MCRNet shows better performance than “MFM-WE”. Besides, the performance of “MFM-WM” is also degraded. This is because that the high-resolution feature contains affluent redundant background noises. In spite of the guidance of the segmentation result, a single 3×3 convolutional layer is not competent to fully eliminate the background noises. Besides, the receptive field size of the 3×3 is very small. Thus, the model

TABLE 6. Average inference speed and complexity comparison with 8 state-of-the-art models. The inference time and MACs of all models are calculated using 384×384 input images.

	MCRNet	MCRNet+	BGNet	BSANet	C ² FNet	ERRNet	PFNet	PraNet	SINet	SINetV2
GMACs	25.02	26.59	49.80	29.70	15.61	23.86	22.62	15.60	23.11	14.61
Param(M)	50.57	50.87	79.85	32.58	28.41	69.76	46.50	32.55	48.95	26.98
Time(s)	0.0325	0.0472	0.0257	0.0355	0.0265	0.0183	0.0235	0.0253	0.0298	0.0264

**FIGURE 10.** Representative failure cases of the MCRNet.

can hardly capture contextual information, which makes the model vulnerable to background interference and cannot generate high-quality prediction maps, especially when the targets are large in size.

E. FAILURE CASES AND ANALYSES

We illustrate three representative failure cases in Figure 10. In the first category, the model fails to segment the camouflaged target perfectly, as some camouflaged regions are misclassified as background areas. This misclassification occurs because the texture of these regions differs significantly from the main body of the targets. Consequently, the model tends to interpret these regions as background noise. The second situation involves the model incorrectly identifying the background as the camouflaged part. This misidentification arises from the relative nature of “camouflage,” where the model’s performance is influenced by the ratio between the size of receptive fields and the scale of actual objects. Additionally, the limited effective receptive field size of CNN-based methods poses challenges. Consequently, when presented with an input image featuring a cluttered background, the model struggles to capture sufficient contextual information to precisely identify the target. The third type of failure case occurs when the model cannot identify targets in dark images.

Generally, most failure cases can be attributed to interference from background information (e.g., contrast, color). We propose ideas aimed at addressing these failure cases. The first suggestion is to use the Vision Transformer [57], [58] as the backbone encoder. Recently, owing to advancements in transformers [57], [58], [59], several transformer-based COD models [60], [61], [62] have been proposed, exhibiting significantly improved performance compared to their CNN-based counterparts. By leveraging the vision transformer, we can better capture contextual information and eliminate background noises, effectively addressing cluttered backgrounds.

Inspired by the success of RGB-D SOD [14], [45], [63], [64] and RGB-T SOD [65], [66] methods, the second idea is to introduce depth images or thermal infrared data as auxiliary information to boost performance. This is because SOD is similar to COD, and SOD methods can perform well on COD datasets after being retrained using COD training samples. Thus, introducing these auxiliaries is a feasible solution to improve performance under challenging scenarios.

F. COMPUTATIONAL COMPLEXITY

We compare the computational complexities of MCRNet and 8 high-performance models including BGNet [3], BSANet [23], C²FNet [22], ERRNet [48], PFNet [8], PraNet [9], SINet [1], and SINetV2 [4] to provide a more comprehensive evaluation. Experiments are conducted on a workstation with a single NVIDIA Titan XP GPU. We repeat each experimental case 50 times to avoid the interference of random factors and ensure more reliable results. The experimental results are demonstrated in Table 6. As can be seen from the table, MCRNet can achieve a real-time inference speed of 31FPS when processing 384×384 images. MCRNet and MCRNet+ are relatively slow, which can be partly attributed to the MFM. As pointed out in [32], low-level features are large in size. Hence, employing these features will largely increase the running time.

V. CONCLUSION

In this paper, we present a novel multi-stage coarse-to-fine refinement framework and introduce a deep model for accurate COD. Specifically, we have devised the BLM to exploit informative cues bilaterally, enhancing the single-layer representation capability. Additionally, we introduce the MFFM to integrate features at various levels and uncover complementary cues between concealed targets

and their boundaries. Subsequently, the MFM is developed to aggregate high-resolution input features with prediction maps from the preceding decoder, eliminating redundant background noise while preserving detailed structural information. We conduct extensive experiments on three COD benchmark datasets, and the results demonstrate that the proposed MCRNet outperforms state-of-the-art counterparts. These findings validate the effectiveness of the proposed model.

REFERENCES

- [1] D. Fan, G. Ji, G. Sun, M. Cheng, J. Shen, and L. Shao, "Camouflaged object detection," in *Proc. CVPR*, 2020, pp. 2774–2784.
- [2] J. Xiao, T. Chen, X. Hu, G. Zhang, and S. Wang, "Boundary-guided context-aware network for camouflaged object detection," *Neural Comput. Appl.*, vol. 35, no. 20, pp. 15075–15093, Jul. 2023.
- [3] Y. Sun, S. Wang, C. Chen, and T. Xiang, "Boundary-guided camouflaged object detection," in *Proc. IJCAI*, 2022, pp. 1335–1341.
- [4] 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.
- [5] M. Stevens and S. Merilaita, "Animal camouflage: Current issues and new perspectives," *Phil. Trans. Roy. Soc. B, Biol. Sci.*, vol. 364, no. 1516, pp. 423–427, Feb. 2009.
- [6] X. Xu, M. Zhu, J. Yu, S. Chen, X. Hu, and Y. Yang, "Boundary guidance network for camouflage object detection," *Image Vis. Comput.*, vol. 114, Oct. 2021, Art. no. 104283.
- [7] 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.
- [8] H. Mei, G.-P. Ji, Z. Wei, X. Yang, X. Wei, and D.-P. Fan, "Camouflaged object segmentation with distraction mining," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8768–8777.
- [9] D. Fan, G. Ji, T. Zhou, G. Chen, H. Fu, J. Shen, and L. Shao, "PraNet: Parallel reverse attention network for polyp segmentation," in *Proc. MICCAI*, 2020, pp. 263–273.
- [10] D.-P. Fan, T. Zhou, G.-P. Ji, Y. Zhou, G. Chen, H. Fu, J. Shen, and L. Shao, "Inf-net: Automatic COVID-19 lung infection segmentation from CT images," *IEEE Trans. Med. Imag.*, vol. 39, no. 8, pp. 2626–2637, Aug. 2020.
- [11] N. Zeng, P. Wu, Z. Wang, H. Li, W. Liu, and X. Liu, "A small-sized object detection oriented multi-scale feature fusion approach with application to defect detection," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–14, 2022.
- [12] H. Chu, W. Hsu, N. J. Mitra, D. Cohen-Or, T. Wong, and T. Lee, "Camouflage images," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 1–8, 2010.
- [13] E. Xie, W. Wang, W. Wang, P. Sun, H. Xu, D. Liang, and P. Luo, "Segmenting transparent objects in the wild with transformer," in *Proc. IJCAI*, 2021, pp. 1194–1200.
- [14] T. Chen, J. Xiao, X. Hu, G. Zhang, and S. Wang, "Adaptive fusion network for RGB-D salient object detection," *Neurocomputing*, vol. 522, pp. 152–164, Feb. 2023.
- [15] X. Qin, Z. Zhang, C. Huang, C. Gao, M. Dehghan, and M. Jagersand, "BASNet: Boundary-aware salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7471–7481.
- [16] S. Li, D. Florencio, Y. Zhao, C. Cook, and W. Li, "Foreground detection in camouflaged scenes," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 4247–4251.
- [17] X. Feng, C. Guoying, H. Richang, and G. Jing, "Camouflage texture evaluation using a saliency map," *Multimedia Syst.*, vol. 21, no. 2, pp. 169–175, Mar. 2015.
- [18] A. Tankus and Y. Yeshurun, "Convexity-based visual camouflage breaking," *Comput. Vis. Image Understand.*, vol. 82, no. 3, pp. 208–237, Jun. 2001.
- [19] F. Xue, C. Yong, S. Xu, H. Dong, Y. Luo, and W. Jia, "Camouflage performance analysis and evaluation framework based on features fusion," *Multimedia Tools Appl.*, vol. 75, no. 7, pp. 4065–4082, Apr. 2016.
- [20] J. Y. H. W. Hou and J. Li, "Detection of the mobile object with camouflage color under dynamic background based on optical flow," *Proc. Eng.*, vol. 15, pp. 2201–2205, Jan. 2011.
- [21] 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.
- [22] Y. Sun, G. Chen, T. Zhou, Y. Zhang, and N. Liu, "Context-aware cross-level fusion network for camouflaged object detection," in *Proc. IJCAI*, 2021, pp. 1025–1031.
- [23] H. Zhu, P. Li, H. Xie, X. Yan, D. Liang, D. Chen, M. Wei, and J. Qin, "I can find you! Boundary-guided separated attention network for camouflaged object detection," in *Proc. AAAI*, 2022, pp. 3608–3616.
- [24] H. Li, C.-M. Feng, Y. Xu, T. Zhou, L. Yao, and X. Chang, "Zero-shot camouflaged object detection," *IEEE Trans. Image Process.*, vol. 32, pp. 5126–5137, 2023.
- [25] J. Ren, X. Hu, L. Zhu, X. Xu, Y. Xu, W. Wang, Z. Deng, and P.-A. Heng, "Deep texture-aware features for camouflaged object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 3, pp. 1157–1167, Mar. 2023.
- [26] A. Li, J. Zhang, Y. Lv, B. Liu, T. Zhang, and Y. Dai, "Uncertainty-aware joint salient object and camouflaged object detection," in *Proc. CVPR*, 2021, pp. 10071–10081.
- [27] J. Wei, S. Wang, and Q. Huang, "F³Net: Fusion, feedback and focus for salient object detection," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, pp. 12321–12328.
- [28] S. Chen, X. Tan, B. Wang, H. Lu, X. Hu, and Y. Fu, "Reverse attention-based residual network for salient object detection," *IEEE Trans. Image Process.*, vol. 29, pp. 3763–3776, 2020.
- [29] S. Li, X. Sui, X. Luo, X. Xu, Y. Liu, and R. S. M. Goh, "Medical image segmentation using squeeze-and-expansion transformers," in *Proc. IJCAI*, Z. Zhou, Ed. 2021, pp. 807–815.
- [30] Y. Pang, X. Zhao, L. Zhang, and H. Lu, "Multi-scale interactive network for salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9410–9419.
- [31] T. Chen, X. Hu, J. Xiao, G. Zhang, and S. Wang, "BINet: Bidirectional interactive network for salient object detection," *Neurocomputing*, vol. 465, pp. 490–502, Nov. 2021.
- [32] Z. Wu, L. Su, and Q. Huang, "Cascaded partial decoder for fast and accurate salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3902–3911.
- [33] H. Zhou, X. Xie, J. Lai, Z. Chen, and L. Yang, "Interactive two-stream decoder for accurate and fast saliency detection," in *Proc. CVPR*, 2020, pp. 9138–9147.
- [34] Y. Pan, Y. Chen, Q. Fu, P. Zhang, and X. Xu, "Study on the camouflaged target detection method based on 3D convexity," *Modern Appl. Sci.*, vol. 5, no. 4, p. 152, Aug. 2011.
- [35] I. H. Casado, D. Rowe, M. Mozerov, and J. González, "Improving background subtraction based on a casuistry of colour-motion segmentation problems," in *Proc. Iberian Conf. Pattern Recognit. Image Anal.*, vol. 4478. Berlin, Germany: Springer, 2007, pp. 475–482.
- [36] X. Hu, S. Wang, X. Qin, H. Dai, W. Ren, D. Luo, Y. Tai, and L. Shao, "High-resolution iterative feedback network for camouflaged object detection," in *Proc. AAAI*, 2023, pp. 881–889.
- [37] Y. Lv, J. Zhang, Y. Dai, A. Li, B. Liu, N. Barnes, and D. Fan, "Simultaneously localize, segment and rank the camouflaged objects," in *Proc. CVPR*, 2021, pp. 11591–11601.
- [38] 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.
- [39] K. Wang, H. Bi, Y. Zhang, C. Zhang, Z. Liu, and S. Zheng, "D²C-Net: A dual-branch, dual-guidance and cross-refine network for camouflaged object detection," *IEEE Trans. Ind. Electron.*, vol. 69, no. 5, pp. 5364–5374, May 2022.
- [40] 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, Sep. 2023.
- [41] J. Zhu, X. Zhang, S. Zhang, and J. Liu, "Inferring camouflaged objects by texture-aware interactive guidance network," in *Proc. AAAI*, 2021, pp. 3599–3607.
- [42] T. Chen, J. Xiao, X. Hu, G. Zhang, and S. Wang, "Boundary-guided network for camouflaged object detection," *Knowl.-Based Syst.*, vol. 248, Jul. 2022, Art. no. 108901.
- [43] 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.

- [44] C. He, L. Xu, and Z. Qiu, "Eldnet: Establishment and refinement of edge likelihood distributions for camouflaged object detection," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2022, pp. 621–625.
- [45] Y. Zhai, D.-P. Fan, J. Yang, A. Borji, L. Shao, J. Han, and L. Wang, "Bifurcated backbone strategy for RGB-D salient object detection," *IEEE Trans. Image Process.*, vol. 30, pp. 8727–8742, 2021.
- [46] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. H. S. Torr, "Deeply supervised salient object detection with short connections," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 4, pp. 815–828, Apr. 2019.
- [47] 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. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 3917–3926.
- [48] 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.
- [49] T. Zhao and X. Wu, "Pyramid feature attention network for saliency detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3080–3089.
- [50] Y. Deng, J. Ma, Y. Li, M. Zhang, and L. Wang, "Ternary symmetric fusion network for camouflaged object detection," *Int. J. Speech Technol.*, vol. 53, no. 21, pp. 25216–25231, Nov. 2023.
- [51] 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.
- [52] 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.
- [53] D. Fan, G. Ji, X. Qin, and M. Cheng, "Cognitive vision inspired object segmentation metric and loss function," *SCIENTIA SINICA Informationis*, vol. 51, no. 9, p. 1475, Sep. 2021.
- [54] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.
- [55] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015, pp. 1–15.
- [56] 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.
- [57] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. ICLR*, 2021, pp. 1–22.
- [58] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "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.
- [59] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9992–10002.
- [60] F. Yang, Q. Zhai, X. Li, R. Huang, A. Luo, H. Cheng, and D.-P. Fan, "Uncertainty-guided transformer reasoning for camouflaged object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 4126–4135.
- [61] 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.
- [62] G. Ji, K. Fu, Z. Wu, D. Fan, J. Shen, and L. Shao, "Full-duplex strategy for video object segmentation," in *Proc. ICCV*, 2021, pp. 4902–4913.
- [63] Z. Zhang, Z. Lin, J. Xu, W.-D. Jin, S.-P. Lu, and D.-P. Fan, "Bilateral attention network for RGB-D salient object detection," *IEEE Trans. Image Process.*, vol. 30, pp. 1949–1961, 2021.
- [64] Q. Chen, Z. Liu, Y. Zhang, K. Fu, Q. Zhao, and H. Du, "RGB-D salient object detection via 3D convolutional neural networks," in *Proc. AAAI Conf. Artif. Intell.*, May 2021, pp. 1063–1071. [Online]. Available: <https://ojs.aaai.org/index.php/AAAI/article/view/16191>
- [65] Q. Zhang, N. Huang, L. Yao, D. Zhang, C. Shan, and J. Han, "RGB-T salient object detection via fusing multi-level CNN features," *IEEE Trans. Image Process.*, vol. 29, pp. 3321–3335, 2020.
- [66] Z. Tu, Y. Ma, Z. Li, C. Li, J. Xu, and Y. Liu, "RGBT salient object detection: A large-scale dataset and benchmark," *IEEE Trans. Multimedia*, vol. 25, pp. 4163–4176, 2023.



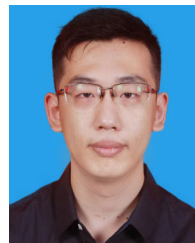
YUYE WANG received the B.S. degree in industrial electrical automation, the M.S. degree in power electronics, and the Ph.D. degree in power electronics and electric drives from Harbin Institute of Technology, Harbin, China, in 1996, 1998, and 2006, respectively. From 1998 to 2006, he was a Lecturer with the School of Electrical Engineering and Automation, Harbin Institute of Technology. From 2006 to 2014, he was an Associate Professor with the College of Information and Communication Engineering, Harbin Engineering University, Harbin. From 2012 to 2013, he was a Visiting Scholar with the School of Science, RMIT University, Melbourne, VIC, Australia. Since 2014, he has been a Professor with the College of Physics and Information Engineering, Minnan Normal University, Zhangzhou, China. He is the author of more than 30 articles and three inventions. His research interests include image processing, nonlinear control, computer-aided testing, and power electronics.



TIANYOU CHEN received the B.S. degree from Huazhong University of Science and Technology, Wuhan, China, in 2015, and the Ph.D. degree from Beihang University in 2024. He is currently an Associate Professor with the Faculty of Artificial Intelligence in Education, Central China Normal University. His current research interests include computer vision and deep learning, especially instance segmentation, salient object detection, and camouflaged object detection.



XIAO GUANG HU (Member, IEEE) received the B.S. degree from Northeast Electric Power University, Jilin, China, in 1983, the M.S. degree from Wuhan University, Wuhan, China, in 1997, and the Ph.D. degree from Harbin Institute of Technology, in 2003. She is currently a Professor with the School of Automation Science and Electrical Engineering, Beihang University. Her research interests include swarm intelligence, embedded test systems, and smart grids.



JIAQI SHI received the B.S. degree from North-western Polytechnical University, in 2020, and the M.S. degree from Beihang University, in 2023, where he is currently pursuing the Ph.D. degree with the School of Automation Science and Electrical Engineering. His research interests include integrated avionics and computer vision.



ZICHONG JIA received the B.S. degree in automation from Beihang University, in 2023, where he is currently pursuing the M.S. degree with the School of Automation Science and Electrical Engineering. His research interests include integrated avionics and computer vision.

• • •